

**Totul
despre**

**Red Hat®
LINUX®**

Naba Barkakati

Traducere de Mihai Mănăstireanu,
Diana Pisai, Bogdan Pisai, Bogdan Mocanu

Teora

Titlul original: Red Hat® LINUX® Secrets 2nd Edition

Copyright © 2002, 2000 Teora

Toate drepturile asupra versiunii în limba română aparțin Editurii Teora.
Reproducerea integrală sau parțială a textului sau a ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al Editurii Teora.

Original English language edition Copyright © 1998 by IDG Books Worldwide, Inc.
All rights reserved including the right of reproduction in whole or in part in any form.
This translation published by arrangement with IDG Books Worldwide, Inc.

Red Hat is a trademark of Red Hat Software, Inc. LINUX is a trademark of Linus Torvalds in the United States and other countries. The IDG Books Worldwide logo is a trademark or registered trademark on the United States and/or other countries under exclusive license to IDG Books Worldwide, Inc., from International Data Group, Inc. Secrets and related trade dress are trademarks or registered trademarks of IDG Books Worldwide, Inc. in the United States and/or other countries. Used by permission.

Teora

Calea Moșilor nr. 211, sector 2, București
fax: 01/210.38.28
e-mail: teora@teora.kappa.ro

Teora – Cartea prin poștă

CP 79-30, cod 72450 București, România
tel: 01/252.14.31
e-mail: cpp@teora.kappa.ro

Copertă: Gheorghe Popescu

Tehnoredactare: Techno Media

Director Editorial: Diana Rotaru

Președinte: Teodor Răducanu

NOT 5097 CAL LINUX RED HAT, TOTUL DESPRE
ISBN 973-20-0291-3

Printed in Romania

CUPRINS

Prefață	7
Introducere	12
Partea I: Configurarea sistemului Linux	14
Capitolul 1: Instalarea sistemului Linux	14
Capitolul 2: Actualizarea sistemului Linux	91
Partea a II-a: Rularea sistemului Linux	125
Capitolul 3: O privire generală asupra sistemului Linux	125
Capitolul 4: Secretele sistemului X sub Linux	139
Capitolul 5: Personalizarea lansării sistemului Linux	168
Capitolul 6: Secretele comenzilor Linux	191
Capitolul 7: Secretele sistemului DOS sub Linux	231
Capitolul 8: Programarea sub Linux în Tcl/Tk	246
Partea a III-a: Exploatarea echipamentelor hardware sub Linux	279
Capitolul 9: Calculatoare	279
Capitolul 10: Plăci video și monitoare	294
Capitolul 11: Unități de disc	312
Capitolul 12: Unități CD-ROM și plăci de sunet	335
Capitolul 13: Tastaturi și dispozitive de indicare	357
Capitolul 14: Imprimante	372
Capitolul 15: Modemuri	399
Capitolul 16: Rețele	426
Capitolul 17: PC Cards	464
Partea a IV-a: Folosirea sistemului Linux pentru distracții și profit.	470
Capitolul 18: Rețele dial-up în Linux	470
Capitolul 19: Configurarea unui calculator gazdă Linux pentru Internet	497
Capitolul 20: Rularea unui server World Wide Web în Linux	518
Capitolul 21: Administrarea unei afaceri cu Linux	539
Capitolul 22: Programarea în Linux	557
Capitolul 23: Programarea X în Linux	591
Capitolul 24: Procesarea textului în Linux	640

Anexe	667
Anexa A: Breviar de aplicații Linux	667
Anexa B: Comenzi Linux	691
Anexa C: Resurse Linux	721
Anexa D: Despre CD-ROM	723
Index	725

Prefață

Dacă sunteți un începător în utilizarea sistemului de operare Linux, aveți nevoie de un ghid practic, care să vă prezinte nu numai instalarea și configurarea acestui sistem, dar și modalitatea de utilizare a lui pentru anumite operații, de exemplu ca gazdă pentru Internet sau ca platformă de dezvoltare software.

Sistemul Linux se află într-o continuă dezvoltare. Nucleul Linux, sistemul de operare de bază, se află acum la versiunea 2.0. Continuă să se dezvolte noi caracteristici ale nucleului, precum și suportul pentru noile echipamente hardware. Interesul utilizatorilor a crescut enorm, așa cum reflectă numărul în continuă creștere al cărților de Linux, destinate cu precădere începătorilor. De asemenea, au fost dezvoltate noi distribuții Linux – combinații ale sistemului de operare cu instrumente de instalare și aplicații – în vederea simplificării procesului de instalare.

În particular, distribuția Linux Red Hat a apărut ca opțiune pentru utilizatorii care își doreau un proces de instalare mai simplu decât cel oferit de Linux Slackware. Red Hat include un număr de instrumente grafice care rulează sub X Window System pentru a asista utilizatorii în operațiile de instalare și configurare. În plus față de deosebirile de instalare, Linux Red Hat diferă de Linux Slackware prin locația și formatul anumitor fișiere de configurare a sistemului. Deși există controverse între specialiști cu privire la supremația unuia sau altuia din sistemele de aranjare a fișierelor, etapele de instalare mai simple permise de distribuția Linux Red Hat sunt categoric în avantajul utilizatorului începător. În orice caz, miezul sistemului de operare Linux și colecția de aplicații sunt aproape identice în aproape toate distribuțiile Linux.

Cartea de față include cea mai recentă versiune de Linux (versiunea 5.1, cu nucleul 2.0.34) pe CD-ROM-ul care însoțește cartea. Această carte include informații tehnice detaliate referitoare la instalarea și personalizarea programului, dar și descrierea diferitelor tipuri de calculatoare și echipamente periferice.

Apoi, această carte depășește cu un pas cadrul literaturii existente în domeniu și prezintă cititorului modul de utilizare a sistemului Linux ca o soluție a anumitor probleme. De asemenea, furnizează informații referitoare la pachete software disponibile gratuit – cum ar fi utilitare de știri, mesaje poștale, grafică și text – pe CD-ROM-ul anexat cărții.

Aspectele unice ale cărții de față sunt sugestiile, tehnicile, scurtăturile și realitățile mai puțin cunoscute referitoare la utilizarea sistemului Linux în diferite operații din lumea reală, de la simpla învățare a sistemului UNIX la configurarea unui server WWW pentru firma dumneavoastră.

Prin lectura acestei cărți veți beneficia de următoarele avantaje:

- Învățați să instalați și să configurați Linux din CD-ROM-ul Linux Red Hat inclus în volumul de față
- Învățați cum să folosiți diferite periferice (plăci video, hard-discuri și plăci de rețea) în Linux.
- Învățați despre legarea în rețea prin apel telefonic (folosind SLIP și PPP) din Linux
- Aflați sugestii, tehnici și scurtături pentru utilizări specifice ale sistemului Linux, cum sunt:
 - Folosirea sistemului Linux ca gazdă Internet (server WWW și server FTP anonim)
 - Modul de coexistență a sistemelor DOS și Linux

- Utilizarea sistemului UNIX pe platforme Linux
- Programarea în C și C++ sub Linux
- Programarea Motif în Linux
- Primiți multe instrumente și utilitare Linux
- Învățați despre resurse Linux care pot servi ca surse continue de informație privind lumea în permanentă schimbare a Linux-ului.

Structura cărții

Cartea de față are 24 de capitole, organizate în patru părți și patru anexe:

Partea I: Configurarea sistemului Linux include două capitole care conțin îndrumările esențiale pentru parcurgerea etapelor de instalare a sistemului Linux de pe CD-ROM-ul care însoțește cartea. Al doilea capitol al acestei părți descrie modul de aplicare a corecțiilor (patches) și de reconfigurare a nucleului (kernel) Linux la apariția de noi versiuni revizuite.

Partea a II-a: Rularea sistemului Linux se concentrează asupra modului de rulare a sistemului Linux după instalare. Această parte include șase capitole care explică modul de configurare a interfeței grafice cu X, modul de utilizare a comenzilor Linux (care, în esență, sunt comenzi UNIX), modul de accesare a sistemului DOS din Linux și modul de utilizare a limbajului de scripting Tcl/Tk pentru o programare rapidă și mai puțin ortodoxă.

Partea a III-a: Exploatarea echipamentelor hardware sub Linux vă pune la dispoziție toate informațiile necesare pentru a folosi diferite tipuri de componente hardware sub Linux. Descrierile echipamentelor acoperă toată gama de componente existente, de la procesorul calculatorului, memorie și magistrală la porturi seriale și plăcile PC (care folosesc interfața PCMCIA).

Partea a IV-a: Folosirea sistemului Linux pentru distracții și profit conține șapte capitole care prezintă modul de utilizare a sistemului Linux în anumite scopuri, cum ar fi configurarea unei gazde pentru Internet, rularea unui server World Wide Web, dezvoltarea de aplicații software sau pregătirea documentelor.

Anexa A: Breviar de aplicații Linux descrie mai multe aplicații Linux frecvent folosite, existente pe CD-ROM-ul care însoțește cartea.

Anexa B: Comenzi Linux prezintă o referință în ordine alfabetică a celor mai importante comenzi Linux.

Anexa C: Resurse Linux prezintă acele resurse de pe Internet unde puteți obține cele mai recente informații despre Linux.

Anexa D: Despre CD-ROM prezintă pe scurt modul de instalare a sistemului Linux de pe CD-ROM-ul anexat cărții.

Dacă sunteți un utilizator începător, trebuie să începeți cu partea I, mai exact cu instalarea sistemului Linux de pe CD-ROM. Dacă ați instalat deja Linux, puteți începe de la capitolul 3, de unde veți învăța cum să utilizați zi de zi Linux cu maximum de randament. Dacă aveți întrebări legate de hardware, citiți direct capitolul dedicat acestuia din partea a III-a. Partea a IV-a descrie o anumită categorie de utilizări Linux. Citiți capitolele dedicate din partea a IV-a pentru a afla care este modul de efectuare a anumitor operații. Când aveți nevoie de informații privind o anumită comandă Linux, treceți la anexa B și căutați comanda respectivă în referința aranjată în ordine alfabetică.

Convenții folosite în această carte

Cartea folosește un stil de notații simplu. Toate listingurile, numele de fișiere, numele de variabile și cuvintele cheie sunt scrise folosind un font monospațiat, pentru simplitatea lecturii. Primele apariții ale unor termeni și concepte noi sunt marcate cu caractere *cursive*. Textul care trebuie introdus de utilizator este scris cu caractere **aldine**.

Fiecare capitol începe cu o scurtă listă a tuturor lucrurilor interesante pe care le veți învăța citindu-l. Sumarul de la sfârșitul capitolului conține ceva mai multe detalii despre subiectele tratate în capitolul respectiv.

Am folosit pictograme pentru a scoate în evidență cu rapiditate informațiile utile. Iată semnificația pictogramelor:



Observație

Pictograma Observație semnalează un fapt interesant, ceva ce am crezut că v-ar plăcea să știți.



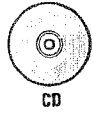
Sugestie

Pictograma Sugestie indică aspecte care vă pot ușura viața — diverse indicii pe care le puteți încerca.



Atenție!

Pictograma Atenție evidențiază posibile pericole. Această ilustrație vă avertizează: Atenție! Vă puteți deteriora sistemul!



CD

Pictograma CD indică anumite programe sau documentație existentă pe CD-ul care însoțește cartea.



Referință încrucișată

Pictograma Referință încrucișată semnalează paragrafe care conduc la un alt capitol din carte, unde există o discuție mai aprofundată a subiectului în cauză.



Secret

Pictograma Secret indică aspecte care nu sunt bine documentate, dar pe care este bine să le cunoașteți. O dată cunoscute, acestea pot clarifica multe probleme.



Vrajitor

Pictograma Vrajitor indică informații tehnice, de interes pentru utilizatorii avansați.

Texte suplimentare

Acesta este un text suplimentar. Am folosit aceste texte pe parcursul cărții pentru a evidenția informații interesante, dar nu esențiale. Textele suplimentare explică concepte pe care probabil că nu le-ați mai întâlnit sau oferă o imagine mai profundă asupra unui subiect co-

relat. Dacă vă grăbiți, puteți omite aceste texte suplimentare fără nici o problemă. Pe de altă parte, dacă vă treziți răsfoind cartea în căutarea unor informații interesante, lectura textelor suplimentare poate fi o idee bună.

CD-ROM-ul Linux Red Hat 5.1

Această carte se adresează utilizatorilor începători care doresc să folosească Linux în scopuri productive pe calculatorul de acasă sau de la birou. Deoarece instalarea este una dintre etapele dificile pe drumul utilizării sistemului Linux, echipa editorială și cu mine am decis să includem o copie a versiunii Linux Red Hat 5.1 (cu nucleul Linux 2.0.34) pe CD-ul anexat cărții. Linux Red Hat este ușor de instalat și are un suport adecvat realizat de Red Hat (www.redhat.com).

Linux Red Hat 5.1 este o distribuție completă Linux, cu nucleul (sistemul de operare) 2.0.34, plus o gamă variată de programe Linux. În particular, iată programele incluse pe CD-ROM-ul atașat cărții:

- Nucleul Linux 2.0.34 cu module driver pentru toate principalele configurații de PC, în speță unități IDE/EIDE și SCSI, dispozitive PCMCIA și unități CD-ROM.
- Set complet de instrumente de instalare și configurare pentru implementarea dispozitivelor periferice (cum ar fi tastatură și mouse) și a serviciilor (rețea)
- Interfață grafică utilizator, bazată pe pachetul XFree86, versiunea 3.3.2, cu utilitarele de gestiune a ferestrelor fvwm2 și AfterStep
- Suport de rețea complet TCP/IP pentru Internet, LAN-uri și intraneturi
- Instrumente pentru conectarea PC-ului dumneavoastră la furnizorii de servicii Internet prin intermediul PPP, SLIP sau al programelor de comunicație serială prin apel telefonic
- Suită completă de aplicații Internet, care conține programe de poștă electronică (sendmail, elm, pine, mailx), de știri (inn, tin, tm), Internet Relay Chat (ircii), telnet, FTP și NFS
- Apache Web Server 1.2.6 pentru transformarea calculatorului dumneavoastră într-un server Web și Netscape Communicator 4.05 pentru a naviga pe Internet
- Samba 1.9 LAN Manager, pentru conectivitatea cu Microsoft Windows
- Multiple editoare de text (GNU Emacs 20.2, JED, Joe, vim)
- Programe de grafică și de manipulare a imaginii, cum ar fi XV, XPaint, Xfig, Gnuplot, Ghostscript, Ghostview, GIMP, ImageMagick și xanim
- Limbaje de programare (GNU C și C++ 2.7.2.3, Perl 5.004, Tcl/Tk 8.0.2, Python 1.5.1, GNU AWK 3.0.3) și instrumente de dezvoltare software (GNU Debugger 4.17, RCS 5.7, GNU Bison 1.25, flex 2.5.4a, biblioteci TIFF și JPEG)
- Suport pentru standardele industriale Executable and Linking Format (ELF) și Intel Binary Compatibility Specification (iBCS)
- Suită completă de utilitare UNIX standard
- Instrumente pentru accesarea și utilizarea fișierelor și aplicațiilor DOS (DOSEMU 0.66.7, mtools 3.8)
- Programe de culegere și formatare a textului (groff, TeX și LaTeX)
- Jocuri precum GNU Chess, xtetris, acm, colour-yantzee, flying, fortune-mod, mysterious, paradise, xchomp, xevil, xgalaga, xgammon, xpilot

Această listă lungă de programe nu trebuie să vă copleșească. Trebuie să învățați să folosiți numai ceea ce vă trebuie. În paralel, această carte vă prezintă atât modul de instalare a sistemului Linux, cât și modul de utilizare a majorității programelor care folosesc platforma acestuia.

Dacă aveți suficient spațiu disponibil pe hard-discul calculatorului dumneavoastră (sau, mai bine, un al doilea hard-disc de rezervă), instalarea sistemului Linux Red Hat poate fi la fel de simplă ca și crearea unei dischete de boot, încărcarea sistemului de operare și furnizarea unor informații într-o serie de casete de dialog. Nu e nevoie să mă credeți pe cuvânt – vă veți convinge singur!

Acum, este momentul să ne începem aventura Linux. Extrageți CD-ROM-ul asociat cărții, treceți la capitolul 1 și... să înceapă distracția. Până să vă dezmeticiți, veți deveni un expert Linux!

Introducere

Dacă ne gândim la apariția și la evoluția sa, Linux este de-a dreptul uluitor. De la începuturile sale modeste, ca hobby al unei singure persoane – finlandezul Linus Torvalds – Linux a evoluat într-un sistem complet operațional pe 32 de biți (64 de biți pe procesorul DEC Alpha), cu caracteristici care rivalizează cu acelea ale sistemelor de operare UNIX x86 comerciale, cum sunt Solaris și SCO UNIX. În fine, „pentru a pune capac la toate“, Linux – cu întreg codul său sursă – este disponibil gratuit pentru oricine. Tot ce aveți de făcut este să îl transferați dintr-un sit Internet sau să-l preluați de pe un CD-ROM, contra unei taxe nominale, de la unul dintre numeroșii distribuitori de UNIX pe CD-ROM.

În mod categoric, Linux este o excepție de la regula „ce plătești, aceea primești“. Chiar dacă Linux este gratuit, nu face absolut nici un rabat de la performanță, caracteristici și fiabilitate. Robustetea sistemului Linux este direct corelată cu modul de dezvoltare a acestuia. Mulți dezvoltatori din toată lumea au colaborat via Internet pentru a adăuga alte și alte caracteristici. Versiunile incrementale sunt transferate în mod continuu de către utilizatori și sunt testate într-o varietate de configurații de sistem. Variantele revizuite Linux trec printr-o testare beta cu mult mai riguroasă decât a oricărui alt program comercial.

De la prima lansare a versiunii Linux 1.0, la 14 martie 1994, numărul de utilizatori Linux din toată lumea a crescut exponențial. Pornind de la datele centralizate de Linux Counter (<http://counter.li.org/>), baza instalată estimată este evaluată între 1 și 5 milioane de utilizatori de pe toată planeta. Aceeași sursă furnizează și alte informații interesante despre Linux:

- Majoritatea utilizatorilor Linux se află în Statele Unite, Canada și Europa.
- Procentajul utilizatorilor Linux care folosesc programul la domiciliu este de 87%, în timp ce 35% îl folosesc la birou; unii îl folosesc și acasă, și la serviciu.
- Procentajul utilizatorilor care preiau Linux via FTP este de 42%. Aproximativ 40% dintre utilizatori cumpără Linux pe CD-ROM, de regulă de la un distribuitor precum InfoMagic sau Red Hat Linux.
- Slackware Linux continuă să fie cea mai populară distribuție Linux; celelalte două distribuții importante sunt Red Hat și Debian.
- Procentajul celor care rulează Linux pe sisteme Intel 486 este 12%, iar 30% rulează sistemul pe calculatoare echipate cu procesoare Pentium. De regulă, sistemele pe care se rulează Linux dispun de un volum de memorie cuprins între 8 și 32 MB RAM, în timp ce spațiul pe disc se situează între 500 MB și 1,5 GB.
- Procentajul sistemelor Linux care folosesc o rețea Ethernet este de 50%, în timp ce 34% folosesc conectarea în rețea prin apel telefonic, prin intermediul protocoalelor SLIP și PPP.
- Majoritatea sistemelor Linux au un număr de utilizatori cuprins între 1 și 8.
- Procentajul sistemelor Linux folosite ca servere Internet (World Wide Web, FTP, Mail, parafo și chiar ca router) este de 61%.

Distribuțiile Linux – Red Hat, Slackware și Debian – diferă sub aspectul procesului de instalare. Deoarece instalarea reprezintă etapa cea mai dificilă la începutul lucrului cu Linux, este utilă selectarea unei distribuții care simplifică procesul de instalare. Linux Red Hat excelează în materie de proces de instalare. Red Hat asigură un program de instalare care ghidează utilizatorii începători printre componentele esențiale ale procesului de instalare,

inclusiv partiționarea discului. Simplitatea instalării transformă Linux Red Hat într-o opțiune recomandată pentru toți utilizatorii, dar mai ales pentru utilizatorii începători.

Linux Red Hat conține un mare volum de informații online pe subiecte precum instalarea și configurarea sistemului de operare pentru o gamă diversă de PC-uri și echipamente periferice. Deși utilizatorii experți pot reuși să instaleze și să ruleze Linux bazați numai pe documentația online, pentru utilizatorii începători este utilă consultarea unei cărți (cum este aceasta) pentru a obține o asistență detaliată cu privire la instalarea Linux. De obicei, utilizatorii trec la Linux cu un anumit scop (cum ar fi configurarea unui server World Wide Web sau pentru a învăța programarea X).

CONFIGURAREA SISTEMULUI LINUX

- Capitolul 1: Instalarea sistemului Linux
- Capitolul 2: Actualizarea sistemului Linux

Capitolul 1 Instalarea sistemului Linux

În acest capitol:

- Să înțelegem complet operația de instalare a sistemului Red Hat Linux de pe CD-ROM-ul care însoțește lucrarea.
- Ce trebuie să știm despre hardware înainte de a instala sistemul Linux
- Primii pași ai instalării sistemului Linux din MS-DOS sau din Windows
- Instalarea sistemului Red Hat Linux de pe CD-ROM-ul atașat cărții
- O primă întâlnire cu sistemul Red Hat Linux

Începând cu apariția procesorului Intel 80386, continuând cu 80486 și în prezent cu Pentium și Pentium II, puterea de calcul a microprocesoarelor încorporate în calculatoarele PC a crescut în mod constant. Viteza ceasului procesoarelor a crescut de la 16 MHz pentru procesorul 80386 acum câțiva ani până la 450 MHz la procesoarele Pentium II. Magistralele ISA (Industry Standard Architecture), apărute încă de la primele calculatoare IBM PC-AT, au fost înlocuite de noile magistrale ultraperformante PCI (Peripheral Component Interconnect). Performanțele hardware ale unui PC modern abia ieșit din linia de fabricație sunt comparabile cu cele ale stațiilor de lucru performante, cum ar fi cele produse de firmele Hewlett-Packard sau Sun.

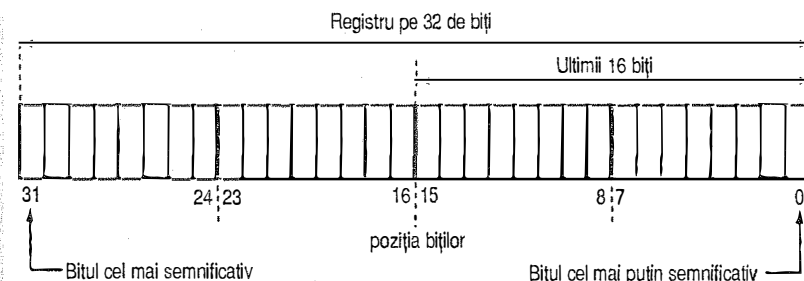
În ceea ce privește însă sistemele de operare, calculatoarele PC nu au ținut pasul cu stațiile de lucru. Multe calculatoare PC rulează încă sisteme de operare pe 16 biți, cum ar fi MS-DOS sau Windows 3.1, care nu exploatează integral capacitatea de prelucrare pe 32 de biți a procesoarelor moderne. Stațiile de lucru, pe de altă parte, rulează UNIX – un sistem de operare multitasking și multiutilizator (aceasta înseamnă că sistemul poate rula simultan mai multe programe și poate permite activitatea mai multor utilizatori). De obicei, stațiile de lucru utilizează mediul X Window System ca interfață grafică pentru utilizatori.

Sistemul UNIX este disponibil în variante pentru PC-uri de ceva vreme, dar configurarea unui sistem de operare UNIX comercial pe un calculator PC costa aproape la fel de mult cât calculatorul însuși. Situația s-a modificat din momentul în care Linus Torvalds de la Universitatea din Helsinki, Finlanda s-a hotărât să construiască un sistem de operare asemănător cu UNIX pentru calculatoarele PC. Ceea ce a început cu un exemplu simplu de

comutare de taskuri cu două procese care afișau AAAA..., respectiv BBBB... pe un terminal alfanumeric s-a dezvoltat până la un sistem de operare multitasking și multiutilizator complet, care reprezintă un concurent serios pentru sistemele UNIX comerciale disponibile pe platformele Intel 80x86. Mulți programatori din toată lumea au contribuit cu cod propriu și au lucrat împreună pentru ca sistemul Linux să devină ceea ce este astăzi. După lansarea în martie 1994 a versiunii 1.0, Linux reprezintă o posibilitate demnă de luat în calcul atât pentru entuziaștii sistemelor UNIX, cât și pentru cei interesați de o platformă UNIX ieftină, dedicată unui anumit scop, cum ar fi dezvoltarea de programe sau configurarea unui sit pe Internet.

Diferențe între sistemele de operare pe 16 biți și cele pe 32 de biți

Procesoarele Intel 80386, 80486 și Pentium dispun de registre de 32 de biți, fiind capabile să prelucreze într-un anumit moment date cu lungimea de 32 de biți. Sistemele de operare pe 16 biți însă, cum sunt MS-DOS sau Windows 3.1, utilizează numai ultimii 16 biți ai registrelor pe 32 de biți (vezi figura), lucrând cu date împărțite în bucăți de 16 biți.



Sistemele de operare pe 32 de biți, cum sunt Linux sau Windows NT, exploatează integral capacitatea registrelor, prelucrând date cu lungimea de 32 de biți în fiecare moment. Pentru a exploata pe deplin capacitatea de prelucrare pe 32 de biți oferită de procesor, aveți nevoie de un sistem de operare pe 32 de biți, cum este și Linux.

După ce veți depăși teama incipientă de necunoscut și veți instala sistemul Linux, veți observa cum puteți utiliza Linux pentru a vă transforma calculatorul PC într-o stație de lucru. Cel mai important avantaj este acela că puteți obține sistemul Linux gratuit – este suficient să-l transferați de pe unul din diferitele surse Internet pe care se găsește. Cea mai bună modalitate de a face primul pas – atât pentru începători cât și pentru experți – este de a cumpăra o carte (cum este și aceasta) care este livrată împreună cu CD-ROM-ul de pe care se poate instala sistemul. Acest ghid complet al sistemului Linux începe cu instalarea sistemului și continuă cu activitățile specifice (cum ar fi dezvoltarea de programe sau conectarea la Internet) pe care le puteți efectua cu ajutorul unui calculator PC pe care ați instalat Linux.



Observație

Instalarea sistemului este una dintre activitățile cele mai complicate în Linux, mai ales dacă aveți un calculator compatibil IBM-PC care nu este produs de o firmă renumită. Aveți nevoie de informații specifice despre dispozitivele hardware din configurația calculatorului, referitoare spre exemplu la tipul controllerului de disc, al plăcii video sau al unității de CD-ROM. Sistemul Linux controlează dispozitivele fizice prin intermediul driverelor, deci trebuie să vă asigurați că versiunea curentă de Linux conține

drivele pentru dispozitivele fizice din configurația dumneavoastră. Din cauză că sistemul Linux este gratuit, nu puteți pretinde – sau nu vă puteți aștepta – să conțină suport pentru unele dispozitive specifice. Sistemul Linux se dezvoltă însă continuu prin colaborarea între programatori din toată lumea. Dacă aveți un dispozitiv suficient de comun, este foarte probabil ca deja cineva să fi scris un driver pentru acel dispozitiv. În orice caz, versiunea Red Hat Linux de pe CD-ROM-ul atașat lucrării suportă o varietate suficient de mare de dispozitive fizice.

Ca prim pas în acumularea experienței de lucru cu sistemul Linux, în acest capitol veți învăța să instalați sistemul de pe CD-ROM-ul atașat. Capitolul începe cu o prezentare generală a întregului proces de instalare; în continuare, procesul este prezentat pas cu pas.



Referință
încrucișată

Dacă ați instalat deja sistemul Linux, puteți afla din capitolul 2 cum să configurați și să actualizați sistemul pentru a beneficia de avantajele sau îmbunătățirile apărute.

Înțelegerea procesului de instalare a sistemului Linux

Înainte de a începe o activitate de durată, este întotdeauna util să evidențiem întreaga secvență de activități pe care urmează să le efectuăm. La fel, înainte de a porni într-o călătorie în locuri în care nu am mai fost, este bine să studiem o hartă a acelor locuri. Instalarea sistemului Linux poate fi o activitate de durată, mai ales dacă ne împiedicăm pe parcurs de anumite obstacole. Această secțiune are rolul de a vă prezenta o hartă generală a procesului de instalare. După ce veți parcurge această secțiune, veți fi pregătit din punct de vedere psihologic pentru a instala sistemul Linux.

Pașii pe care îi veți parcurge efectiv la instalarea sistemului Linux depind de *distribuția* Linux – adică de sursa precisă a sistemului de operare – pe care o veți utiliza. Această carte vă prezintă pașii de instalare de pe CD-ROM-ul atașat. Iată care sunt pașii principali la instalarea sistemului Red Hat Linux:

1. Culegeți informații despre dispozitivele hardware care intră în alcătuirea PC-ului dumneavoastră înainte de a instala sistemul Linux. Sistemul Linux gestionează și utilizează diferitele dispozitive periferice ale PC-ului prin intermediul unor componente software numite *drivele*. Trebuie să vă asigurați că versiunea sistemului Linux pe care o veți instala dispune de driverele necesare pentru configurația hardware a calculatorului dumneavoastră. Dacă încă nu aveți un calculator, aruncați o privire prin lista dispozitivelor hardware suportate de sistemul Linux și asigurați-vă că achiziționați un PC care cuprinde numai componente cunoscute.
2. Este necesar să efectuați o activitate cunoscută sub numele de *partiționare* pentru a alocă anumite părți ale discului dumneavoastră spre a fi utilizate de sistemul Linux. Dacă aveți norocul de a dispune de un al doilea hard-disc complet liber, puteți decide să păstrați MS-DOS și Windows pe primul hard-disc și să instalați sistemul Linux pe cel de-al doilea. Cu un al doilea disc liber, scăpați de grija de a repartitiona discul în DOS sau Windows. Dacă dispuneți însă de un singur disc, trebuie să-l împărțiți în mai multe partiții. Utilizați una dintre partiții pentru DOS și Windows, iar restul lăsați-l pentru Linux.
3. În DOS sau Windows, creați o dischetă de pornire (boot) pentru Linux (sau o simplă dischetă de pornire). Discheta de pornire este utilizată pentru a porni calculatorul și pentru a încărca o versiune inițială a sistemului de operare Linux. Dacă porniți PC-ul în sistemul MS-DOS (este necesar să fie un sistem MS-DOS și nu o fereastră MS-DOS din Windows 95), puteți sări peste acest pas încărcând sistemul Linux direct de pe CD-ROM.

4. Porniți PC-ul cu discheta de boot Linux (sau porniți sistemul MS-DOS și executați comanda de încărcare a CD-ROM-ului). Această procedură rulează automat programul de instalare a sistemului Red Hat Linux. De aici încolo, va fi necesar să răspundeți la un număr de casete de dialog pe măsură ce programul de instalare a sistemului Red Hat vă poartă pas cu pas prin procesul de instalare.

5. Răspundeți la caseta de dialog care vă întreabă dacă aveți un monitor color. În casetele de dialog ulterioare, veți selecta tipul tastaturii și veți indica sursa sistemului Red Hat Linux (unitatea CD-ROM).

6. Pregătiți partițiile hard-discului pe care ați planificat să instalați sistemul Linux. După ce ați creat spațiu pentru sistemul Linux prin reducerea partiției DOS existente, este rândul să creați partițiile pentru Linux. De obicei, avem nevoie de cel puțin două partiții: una pentru fișierele sistemului Linux și cealaltă pentru a fi utilizată ca *partiție de swap*, o formă de memorie virtuală. Pentru a efectua acest pas, programul de instalare vă permite să optați între utilizarea comenzii Linux `fdisk` și a noului utilitar Red Hat de gestiune a hard-discului denumit *Disk Druid*.

7. Formatați partițiile hard-discului, indicați care dintre partiții este cea de swap și specificați partiția pe care se va instala sistemul Linux (numită și partiție rădăcină). Puteți de asemenea să montați și alte tipuri de partiții. Montarea unei partiții determină asocierea dintre o partiție fizică de pe disc și un director din sistemul de fișiere Linux.

8. Selectați diferitele componente software pe care le veți instala. Fiecare dintre aceste componente reprezintă o parte a sistemului Linux – de la sistemul de operare de bază și până la componente cum ar fi editorul `emacs`, instrumente de dezvoltare de programe, limbajul Perl sau sistemul de ferestre X (un sistem de ferestre în mod grafic). Este suficient să selectați componentele de care aveți nevoie și programul de instalare al sistemului Red Hat își va face conștiincios datoria.

9. Dacă ați instalat sistemul X, programul de instalare va configura mouse-ul și va rula programul `Xconfigurator`, care creează un fișier de configurare (`/etc/X11/XF86Config`) utilizat de sistemul X. Ca răspuns la casetele de dialog prezentate de programul `Xconfigurator`, veți introduce informații despre placa video și monitorul dumneavoastră.

10. Dacă ați instalat software pentru rețea, programul de instalare vă permite să configurați rețeaua. Va trebui să precizați un număr de parametri, printre care adresa IP, numele gazdei dumneavoastră și numele domeniului din care PC-ul dumneavoastră va face parte.

11. Precizați fusul orar local.

12. Configurați toate imprimantele de care dispuneți. Puteți configura atât o imprimantă direct conectată la PC-ul dumneavoastră, cât și o imprimantă aflată în rețea. Sistemul Red Hat vă permite să configurați chiar și o imprimantă conectată la un PC care rulează Windows 95 sau Windows NT.

13. Alegeți o parolă de root. Utilizatorul „root” este *superutilizatorul* – cel care are dreptul de a face orice – în sistemul Linux.

14. Instalați programul Linux Loader (LILO) pe discul dumneavoastră, astfel încât să puteți încărca sistemul Linux la fiecare pornire a PC-ului, după ce acesta a fost oprit.

15. Dacă observați că sub Linux una sau mai multe componente ale PC-ului dumneavoastră (unitatea de CD-ROM, placa de sunet etc.) nu funcționează corect, va trebui să reconfigurați sistemul Linux, adăugând suport pentru componentele respective.

Secțiunile următoare vă vor ghida în ceea ce privește pașii procesului de instalare și pornirea inițială a sistemului Linux.



Observație

PC-ul dumneavoastră trebuie să dispună de o unitate CD-ROM – dintre cele pe care sistemul Linux le cunoaște – pentru a putea efectua instalarea de pe CD-ROM-ul atașat acestei cărți. Dacă PC-ul dumneavoastră nu dispune de unitate CD-ROM, dar este legat în rețea, puteți utiliza unitatea CD-ROM a unui alt PC pentru a instala Linux din rețea folosind NFS sau FTP.

Cum pregătim un PC pentru instalarea sistemului Linux

Înainte de a instala Linux, trebuie să vă pregătiți calculatorul pentru instalare. În acest moment, vă puteți afla în una din următoarele două situații:

- Dispuneți de un PC care rulează unul din sistemele de operare tradiționale pentru PC-uri, cum ar fi MS-DOS, Microsoft Windows 3.1, Windows 95, Windows NT sau OS/2.
- Sunteți pe punctul de a cumpăra un PC și doriți să rulați Linux pe acel PC cel puțin din când în când.

Dacă sunteți pe cale de a cumpăra un PC, vă aflați într-un caz fericit, deoarece puteți cumpăra un PC care să cuprindă în configurație numai periferice cunoscute de către Linux. Pentru a alege o configurație compatibilă cu sistemul Linux, tot ce aveți de făcut este să consultați lista curentă de dispozitive suportate de Linux și apoi să selectați un PC care include numai componente hardware suportate. Este necesar să cereți explicit celui de la care cumpărați calculatorul informații detaliate referitoare la periferice cum ar fi placa video, unitatea CD-ROM sau placa de rețea, pentru a vă asigura că veți putea utiliza aceste periferice sub Linux. Alegerea unui PC cu dispozitive suportate de Linux reduce considerabil potențialul apariției unor probleme la instalarea sistemului Linux. Următoarele secțiuni cuprind o listă a dispozitivelor hardware suportate de Linux.

Puteți de asemenea să cumpărați un PC cu sistemul Linux gata instalat. Puteți găsi oferte pentru astfel de stații Linux în revista *Linux Journal* (vezi anexa C).

Dacă doriți să instalați Linux pe un PC existent, verificați dacă ultima distribuție suportă toate componentele hardware ale PC-ului dumneavoastră. Cu alte cuvinte, este necesar să faceți un inventar al componentelor hardware și să stabiliți dacă Linux le suportă pe fiecare dintre ele.

Inventarul componentelor PC-ului dumneavoastră

Ca și multe alte sisteme de operare, Linux gestionează diferitele tipuri de componente fizice prin intermediul driverelor pentru dispozitive. Pentru fiecare tip de dispozitive periferice, cum ar fi plăcile de rețea sau unitățile de CD-ROM, Linux are nevoie de un driver. De fapt, fiecare tip de periferice necesită un driver separat. Din cauză că sistemul Linux este disponibil gratuit (sau la un preț relativ redus) și din cauza faptului că sistemul a fost și continuă să fie dezvoltat prin cooperarea unui număr mare de programatori din toată lumea, nu puteți avea pretenții la suport pentru anumite tipuri speciale de dispozitive fizice. Cea mai mare speranță a dumneavoastră este ca o persoană care știe să scrie un driver pentru Linux să dispună de același dispozitiv ca și dumneavoastră. În acest caz, este foarte probabil ca persoana respectivă să scrie efectiv driverul, iar acesta să își găsească loc într-o anumită versiune a sistemului Linux; astfel, veți putea utiliza la rândul dumneavoastră acel dispozitiv sub Linux.

Verificați lista dispozitivelor suportate de versiunea de Linux de pe CD-ul atașat lucrării (lista va fi trecută în revistă în următoarele câteva secțiuni). Puteți instala și rula sistemul Linux chiar dacă pentru anumite periferice nu există drivere disponibile. Oricum, pentru a putea instala Linux de pe CD-ul atașat, aveți nevoie cel puțin de un procesor compatibil cu sistemul Linux, o magistrală corespunzătoare, un hard-disc, o tastatură și o unitate CD-ROM. Dacă doriți să rulați și sistemul X Window, trebuie de asemenea să vă asigurați că XFree86 (serverul X pentru Linux) suportă mouse-ul, placa video și monitorul. (În capitolul 4, veți afla totul despre X.)

Următoarele secțiuni conțin o trecere în revistă a dispozitivelor fizice suportate de versiunea de Linux de pe CD-ROM-ul atașat.



Referință încrucișată

Începând cu capitolul 9 și terminând cu capitolul 17, sunt prezentate detaliat toate dispozitivele fizice dintr-un PC. Consultați aceste capitole pentru a afla dacă Linux suportă configurația unică de care dispuneți. În aceste capitole, puteți de asemenea găsi informații despre cum vă puteți utiliza resursele hardware ale PC-ului dumneavoastră sub Linux cu maximum de eficiență.

Procesorul

Procesorul este unitatea centrală de prelucrare (CPU) – circuitul integrat care efectuează toate prelucrările într-un PC. Aveți nevoie de cel puțin un procesor Intel 80386 pentru a putea rula Linux. Orice procesor compatibil cu procesoarele Intel 80386, 80486 sau Pentium este capabil de a rula sistemul de operare Linux. Printre procesoarele compatibile, Linux poate rula pe AMD K5 sau K6, cât și pe procesoare Cyrix sau IBM.



Observație

Nu puteți rula Linux pe un PC 80286, deoarece 80286 nu este un procesor pe 32 de biți.

Magistrala

Magistrala reprezintă conexiunea electrică standard dintre procesor și perifericele acestuia. Există mai multe tipuri de magistrale pentru PC-uri. Cea mai cunoscută dintre acestea este magistrala Industry Standard Architecture (ISA), anterior numită *magistrală AT* datorită faptului că firma IBM a utilizat-o pentru prima dată într-un calculator IBM PC-AT în 1984. Alte tipuri de magistrale sunt Extended Industry Standard Architecture (EISA); VESA Local Bus (VLB); Micro Channel Architecture (MCA) și, cea mai recentă, Peripheral Component Interconnect (PCI).



Observație

Linux suportă toate magistralele PC obișnuite. Suportul pentru magistrala MCA (utilizată în calculatoarele IBM PS/2) a apărut începând cu versiunea 2.0.7 a nucleului sistemului Linux (numerele versiunilor nucleului sistemului Linux sunt explicate în capitolul 2).

Memoria

De obicei denumită *memorie cu acces aleator* (Random Access Memory sau RAM), memoria nu reprezintă un factor de compatibilitate. Aveți însă nevoie de cel puțin 8 MB de RAM pentru a obține performanțe cât de cât acceptabile. Deși este posibil să instalați Linux pe un PC cu numai 4 MB de RAM, nu veți putea rula sistemul X Window pe acel PC. Sistemul X Window gestionează interfața grafică prin intermediul unui *server X*, care este un program mare și care are nevoie de multă memorie pentru a putea rula eficient.



Secret

Dacă sunteți în situația de a cumpăra un PC, luați-l cu cel puțin 16 MB de RAM. Dacă aveți un PC mai vechi cu mai puțin de 8 MB de RAM, este bine să mai adăugați memorie pentru a ajunge la cel puțin 16 MB în total. Cu cât un sistem dispune de mai multă memorie fizică, cu atât sistemul va putea rula mai eficient mai multe programe, deoarece programele vor încăpea toate în memorie. Deși Linux poate utiliza o parte a hard-discului ca memorie virtuală, o astfel de memorie, bazată pe utilizarea discului, este cu mult mai lentă decât memoria fizică. Cantitatea de memorie fizică necesară depinde de mărimea sistemului de operare Linux și de mărimea programelor care trebuie să ruleze simultan, cum ar fi sistemul X Window. Deși sistemul Linux singur poate rula în doar 4 MB de memorie, aveți nevoie de cel puțin 8 MB pentru a rula X. Adăugați la această cantitate orice aplicație (cum ar fi un editor sau un compilator) pe care va fi necesar să o rulați și veți realiza repede de ce aveți nevoie de cel puțin 16 MB de RAM pentru obținerea unor performanțe adecvate.

Placa video și monitorul

Majoritatea PC-urilor dispun de plăci video cunoscute sub numele de *Super VGA* (Video Graphics Array). Sistemul Linux lucrează bine cu toate plăcile video în mod text. Când ajungem însă la XFree86 – versiunea pentru Linux a sistemului X Window – lucrurile stau altfel. Dacă XFree86 nu suportă explicit placa video, veți avea serios de lucru pentru a configura XFree86 astfel încât să lucreze adecvat cu placa dumneavoastră.

Tipul monitorului pe care îl utilizați nu este chiar esențial, dar acesta trebuie să fie capabil să afișeze la rezoluțiile pe care placa video le utilizează, măsurate în pixeli, pe orizontală și pe verticală (spre exemplu, 1024x768).

În general, suportul oferit de XFree86 pentru o placă video depinde de circuitul integrat care controlează monitorul și determină modul în care acesta efectuează afișarea; acest circuit este cunoscut sub denumirea de *cipset video*. Producătorii de plăci video pot însă utiliza cipseturile video în manieră nestandard. Într-un astfel de caz, aveți nevoie de o versiune specială a serverului XFree86 pentru a putea lucra cu placa video. Pentru a vă ajuta în alegerea unei plăci video, prezentăm în continuare o listă de plăci video suportate de XFree86 versiunea 3.3.1 (versiune inclusă pe CD-ROM-ul atașat):

- ATI Mach8, ATI Mach32 și ATI Mach64
- seria ATI VGA Wonder
- Compaq AVGA
- DEC 21030
- Diamond Viper VLB și Viper PCI
- Enhanced Graphics Adapter (EGA)
- Genoa GVGA
- Hercules monocrom
- Hyundai HGC-1280 monocrom
- IBM 8514/A, XGA și XGA-II
- Matrox MGA2064W (Millennium) și Matrox MGA1064SG (Mystique)
- Number Nine Imagine I128
- NVidia NVI
- Orchid P9000

- Sigma Laser View PLUS monocrom
- Video 7 (cunoscută și sub denumirea de Headland Technologies HT216-32)
- plăci video bazate pe cipseturile Advance Logic AL2101, 2228, 2301, 2302, 2308 și 2401
- plăci video bazate pe cipseturile Alliance AP6422 și AT24
- plăci video bazate pe cipseturile ARK Logic ARK1000PV, ARK1000VL, ARK2000PV și ARK2000MT
- plăci video bazate pe cipseturile Chips & Technologies 64200, 64300, 65520, 65525, 65530, 65535, 65540, 65545, 65546, 65548, 65550 și 65554
- plăci video bazate pe cipseturile Cirrus Logic CLGD5420, 542x, 5430, 5434, 5436, 544x, 546x, 5480, 62x5, 6420, 6440, 754x și 7555 (x poate fi orice cifră)
- plăci video bazate pe cipseturile IIT AGX-010, 014, 015 sau 016
- plăci video bazate pe cipseturile MX68000 sau MX68010
- plăci video bazate pe cipseturile NCR 77C22, 77C22E și 77C22E+
- plăci video bazate pe cipseturile OAK OTI-067, OTI-077 și OTI-087
- plăci video bazate pe cipsetul RealTek RTG3106
- plăci video bazate pe cipseturile S3 732 (Trio 732), 764 (Trio 764), Trio 64V+, 801, 805, 864, 866, 868, 86C325 (ViRGE), 86C375 (ViRGE/DX), 86C385 (ViRGE/GX), 86C988 (ViRGE/VX), 911, 924, 928, 964 sau 968 (vezi capitolul 4 pentru lista completă a plăcilor S3 suportate)
- plăci video bazate pe cipsetul SGS-Thomson STG2000
- plăci video bazate pe cipseturile SiS 86c201, 86c202 sau 86c205
- plăci video bazate pe cipsetul Weitek P9000
- plăci video bazate pe cipseturile Trident 8800CS, 8200LX, 8900x, 9000, 9000i, 9100B, 9200CXr, 9320LCD, 9400CXi, 9420, 9420DGi, 9440, 96xx și Cyber938 (x poate reprezenta orice cifră)
- plăci video bazate pe cipseturile Tseng ET3000, ET4000, ET4000AX, ET6000, W32, W32i și W32p
- plăci video bazate pe cipseturile Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C31 și WD90C33
- Video Graphics Array (VGA)
- Western Digital Paradise PVGA1

Pentru instalarea componentelor de bază ale sistemului Linux, nu trebuie să vă preocupe placa video. Informațiile detaliate despre placa video devin importante abia atunci când doriți să configurați XFree86.



Referință încrucișată

Deși pe parcursul instalării sistemului Red Hat Linux vom efectua o configurare a serverului X, puteți găsi detalii suplimentare referitoare la configurarea XFree86 în capitolul 4.

Unitatea de hard-disc

Sistemul Linux poate suporta toate unitățile de hard-disc suportate de BIOS-ul (Basic Input and Output System) PC-ului dumneavoastră. La multe PC-uri 386 sau 486 mai vechi, era

nevoie să instalați un driver separat pentru a putea gestiona unități de disc mai mari – BIOS-ul sistemului nu putea să lucreze cu aceste unități. Nu puteți instala Linux pe astfel de sisteme. Pe scurt, sistemul Linux suportă o unitate de disc numai dacă aceasta este suportată de BIOS-ul sistemului, cu o restricție semnificativă. Pentru a putea porni sistemul Linux de pe o unitate de disc mare (cu mai mult de 1024 de cilindri), programul Linux Loader (LILO), nucleul sistemului Linux și fișierele de configurare pentru LILO trebuie să se găsească în primii 1024 de cilindri ai unității. Aceasta deoarece sistemul Linux utilizează BIOS-ul pentru a încărca nucleul, iar BIOS-ul nu poate gestiona cilindri dincolo de primii 1024.

Pentru unitățile de hard-disc conectate la PC-ul dumneavoastră prin intermediul unui controller SCSI, sistemul Linux are nevoie de un driver, care permite controllerului să gestioneze și să utilizeze aceste unități. O listă a controllerelor suportate se găsește în secțiunea „Controllerul SCSI” din acest capitol.



Sugestie

Singura decizie care mai rămâne de luat referitor la alegerea unității de disc este legată de capacitatea acesteia. Dacă aveți un PC mai vechi, este foarte probabil că aveți o unitate de disc de capacitate relativ mică – la nivelul de aproximativ 500 MB.

Deși este posibil să instalați Linux și sistemul X Window într-un spațiu de 400 MB, în acest caz spațiul rămas pentru Windows 95 este insuficient, în caz că doriți să îl păstrați. În concluzie, dacă PC-ul dumneavoastră are o singură interfață IDE și un hard-disc de capacitate redusă (500 MB sau mai puțin), puteți adăuga un al doilea hard-disc, deoarece majoritatea controllerelor IDE suportă două unități de hard-disc. Dacă dispuneți de o placă SCSI, puteți conecta pe aceasta până la 7 dispozitive SCSI.

Dacă tocmai doriți să cumpărați un PC nou, amintiți-vă că o instalare completă a sistemului Linux (cu toate pachetele Red Hat) necesită aproximativ 400 MB de spațiu pe disc. În plus, veți avea nevoie de un spațiu pe disc pentru activitatea dumneavoastră. Din fericire, majoritatea PC-urilor din zilele noastre sunt livrate cu discuri de dimensiuni cuprinse între 1,6 GB și 6 GB. Oricare dintre aceste discuri este suficient de mare pentru a permite menținerea sistemului Windows pe una dintre partiții, cu opțiunea de a porni unul din sistemele Windows sau Linux.

BIOS-ul unui PC

Toate calculatoarele compatibile IBM PC sunt livrate împreună cu un BIOS construit în memoria read-only (ROM). BIOS-ul conține un set de funcții de intrare-ieșire (I/O) necesare pentru accesul la dispozitivele periferice ale PC-ului, cum sunt tastatura, monitorul, porturile seriale și unitățile de hard-disc sau de dischetă.

BIOS-ul este în esență un program memorat în ROM; prin urmare, firmele care comercializează calculatoare revizuiesc și actualizează periodic și BIOS-ul. De obicei, revizuirea BIOS-ului are loc în scopul lucrului cu noi dispozitive, cum ar fi noile unități de disc cu capacitate mult mai mare decât a fost inițial prevăzută, sau noile tipuri de magistrale, ca de exemplu magistralele PCI. Noile versiuni de BIOS pot de asemenea îmbunătăți performanțele

sistemului executând cu mult mai eficient anumite activități. Din cauză că BIOS-ul este un element crucial în ceea ce privește modul în care sistemul Linux lucrează cu perifericele unui PC, este indicat să actualizați BIOS-ul pentru a putea rula Linux pe un calculator PC 386 sau 486. Actualizarea presupune înlocuirea unei perechi de circuite cu versiuni mai noi – pentru a obține informații la zi despre circuitele BIOS compatibile cu PC-ul dumneavoastră, va fi necesar să contactați firma producătoare.

Un alt motiv de a actualiza BIOS-ul PC-ului (și foarte probabil și ceasul de timp real din CMOS) îl constituie problema anului 2000. Aceasta constă în imposibilitatea PC-ului de a memora corect anul curent îndată ce acesta devine egal cu 2000.



Gândiți-vă serios la varianta cumpărării unui al doilea hard-disc pentru Linux. Un al doilea disc elimină multe din riscurile procesului de instalare, din cauză că nu mai trebuie să partiționați discul pe care este instalat Windows 95.

Unitatea de dischete

Ca și pentru hard-disc, driverele sistemului Linux utilizează BIOS-ul PC-ului pentru accesul la dischete. Prin urmare, unitatea de dischete este compatibilă cu sistemul Linux. Se poate chiar să fiți nevoiți să încărcați sistemul Linux de pe o dischetă în procesul de instalare. Pentru aceasta, aveți nevoie de o unitate de dischete high-density de 5,25 inci (cu capacitatea de 1,2 MB) sau de 3,5 inci (cu capacitatea de 1,44 MB). Puteți evita încărcarea de pe dischetă pornind PC-ul în sistemul MS-DOS (nu într-o fereastră MS-DOS din Windows 95) și efectuând un acces la CD-ROM direct de la promptul DOS.

Tastatura și mouse-ul

Sistemul Linux suportă orice tastatură cu care PC-ul dumneavoastră poate lucra în mod curent, dar pentru mouse este nevoie de suport explicit. Aveți neapărat nevoie de mouse dacă doriți să configurați și să rulați XFree86, varianta pentru Linux a sistemului X Window.

Sistemul Linux suportă următoarele tipuri obișnuite de mouse:

- Mouse-ul serial Microsoft
- Mouse-ul serial Mouse Systems
- Mouse-ul serial Logitech Mouseman
- Mouse-ul serial Logitech
- Mouse-ul de magistrală Logitech
- Mouse-ul PS/2
- Mouse-ul de magistrală Microsoft
- Mouse-ul de magistrală ATI XL Inport
- Mouse-ul QuickPort (C&T 82C710), utilizat la PC-urile laptop TI Travelmate sau Toshiba.

Controllerul SCSI

Interfața Small Computer System Interface, denumită în mod uzual SCSI (se pronunță *scazi*), reprezintă modalitatea standard de conectare a multor tipuri de dispozitive periferice la un calculator. Multe tipuri de calculatoare dispun de interfețe SCSI – începând cu stațiile UNIX și până la PC-urile obișnuite. Pentru a utiliza un dispozitiv SCSI pe PC-ul dumneavoastră, este nevoie să conectați o placă de controller SCSI în unul din sloturile magistralei PC-ului dumneavoastră.

De obicei, dispozitivele conectate prin intermediul unui controller SCSI sunt unitățile de hard-disc sau unitățile CD-ROM. Dacă doriți să utilizați un dispozitiv SCSI în Linux, trebuie mai întâi să vă asigurați că sistemul Linux suportă placa de controller SCSI de care dispuneți.

Versiunea sistemului Linux de pe CD-ROM-ul atașat suportă următoarele controllere SCSI:

- Acculogic ISApport, bazat pe cipsetul NCR 53c406a
- Adaptec AHA-1510 și AHA-152x (bazate pe cipurile AIC-6260 sau AIC-6360; x poate fi orice cifră), pentru magistrala ISA
- Adaptec AHA-154x (x se poate înlocui cu orice cifră) pentru magistrala ISA

- Adaptec AHA-174x (x poate fi orice cifră), pentru magistrala EISA (Extended Industry Standard Architecture)
- Adaptec AHA-274x pentru magistrala EISA și AHA-284x pentru magistrala VLB (VESA Local Bus); ambele utilizează cipul AIC-7770
- Adaptec AHA 2920
- Adaptec AHA-2940 și AHA-3940 pentru magistrala PCI; aceste plăci sunt bazate pe cipul AIC-7870
- Adaptec AVA-1505 și AVA-1515 pentru magistrala ISA (suportate de driverul pentru Adaptec 152x în Linux)
- Always IN2000
- AMI Fast Disk VLB/EISA (suportat de driverul Bus Logic în Linux)
- Bus Logic pentru ISA, EISA, VLB și PCI (toate modelele)
- DPT PM2001 și PM2012A
- DPT Smartcache pentru magistralele ISA, EISA și PCI, incluzând modelele PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224 și PM3334
- DTC 329x pentru magistrala EISA (suportat de driverul Adaptec 154x în Linux)
- Future Domain TMC-16x0 și TMC-3260 (PCI)
- Future Domain TMC-8xx și TMC-950
- controllerele PCI ICP-Vortex Disk Array Controllers, incluzând modelele GDT6111RP, GDT6121RP, GDT6117RP, GDT6511RP, GDT6521RP, GDT6517RP, GDT6527RP, GDT6537RP și GDT6557RP
- controllerele EISA ICP-Vortex, care cuprind modelele GDT3000B, GDT3000A, GDT3010A, GDT3020A și GDT3050A
- controllerul Media Vision Pro Audio Spectrum 16 pentru magistrala ISA
- controllerul Media Vision Premium 3D, bazat pe cipsetul NCR 53c406a
- plăcile SCSI generice NCR 5380
- controllerul Qlogic FAS408
- Quantum ISA-200S și ISA-200MG
- controllerele SCSI bazate pe cipseturile NCR 53c7x0 și 53c8x0 (pentru magistrala PCI)
- controllerele Seagate ST-01 și ST-02 pentru magistrala ISA
- controllerul Sound Blaster 16 pentru magistrala ISA (suportat în Linux prin intermediul driverului Adaptec 152x)
- Tekram DC-390, DC-390W/U/F
- controllerele Trantor T128, T128F și T228 pentru magistrala ISA
- Trantor T130B, bazat pe cipsetul NCR 53c400 (suportat cu ajutorul driverului pentru NCR 5380)
- UltraStor 14F (pentru magistrala ISA), UltraStor 24F (pentru magistrala EISA) și UltraStor 34F (pentru VLB)
- controllerele Western Digital WD7000

**Sugestie**

Lista de mai sus va crește pe măsură ce cei care contribuie la dezvoltarea sistemului Linux vor adăuga suport pentru noi controllere SCSI. Pentru o listă mai completă, verificați documentația on-line inclusă pe CD-ROM. Secțiunea „Căutarea prin documentația on-line“ de la sfârșitul acestui capitol prezintă modul în care puteți găsi și folosi această documentație.

Unitatea CD-ROM

Unitățile CD-ROM (Compact Disc Read-Only Memory) sunt extrem de populare din cauză că fiecare CD-ROM poate înmagazina până la 650 MB de informație. Aceasta reprezintă o capacitate mare de memorare comparativ cu cea oferită de o dischetă. CD-ROM-urile sunt fiabile și ieftine din punct de vedere al fabricației. Firmele producătoare utilizează de regulă CD-ROM-urile pentru a distribui o cantitate mare de informație la un preț rezonabil.

Această carte oferă sistemul Linux pe un CD-ROM; prin urmare, pentru a instala software-ul de pe acesta, aveți nevoie de o unitate CD-ROM. Majoritatea PC-urilor noi sunt livrate împreună cu o astfel de unitate. Dacă însă configurația de bază nu include o unitate CD-ROM, puteți să-i adăugați una cu un cost suplimentar egal cu o fracțiune din costul PC-ului – de regulă, suma este aproximativ egală cu echivalentul a 100 \$ S.U.A. Dacă dispuneți de un PC mai vechi care nu are încorporată o unitate CD-ROM, trebuie să achiziționați unitatea pentru a putea instala sistemul Linux de pe CD-ROM-ul atașat acestei lucrări.

Unitățile CD-ROM au devenit populare în lumea PC-urilor în ultimii ani pe măsură ce tot mai mulți utilizatori au optat pentru facilitățile multimedia pe care acestea le oferă. După cum poate știți, în lumea PC-urilor, termenul de *multimedia* desemnează utilizarea unor surse diferite de informație – sunet, imagine, animație și video – în aplicațiile software. Placa de sunet și unitatea CD-ROM reprezintă cele două elemente comune tuturor aplicațiilor multimedia.

Combinăția dintre plăcile de sunet și unitățile CD-ROM a fost până nu demult atât de în vogă încât multe plăci de sunet (cum este și placa Sound Blaster Pro, a firmei Creative Labs) erau livrate însoțite de o unitate CD-ROM. Unitatea CD-ROM era conectată printr-un cablu la placa de sunet, care dispunea de conectorul fizic adecvat. Sistemul Linux suportă unitățile CD-ROM (cum este și Sound Blaster Pro CD) conectate direct la placa de sunet.

În prezent, majoritatea unităților CD-ROM utilizează interfața Enhanced Integrated Drive Electronics (EIDE), care realizează de asemenea conectarea unităților de hard-disc la PC. Sistemul Linux suportă de asemenea unitățile CD-ROM EIDE.

Unele unități CD-ROM sunt dispozitive SCSI, conectate la o placă de controller SCSI. Sistemul Linux suportă unitățile CD-ROM SCSI dacă dispune de driver pentru controllerul SCSI asociat.

În continuare, iată câteva dintre unitățile CD-ROM suportate de Linux:

- Orice unitate CD-ROM EIDE (referită de asemenea prin denumirea AT Attachment Packet Interface, sau unitate ATAPI)
- Orice unitate SCSI care poate transfera informația în blocuri de 512 sau de 2048 de octeți (acest standard include majoritatea unităților CD-ROM disponibile pe piață)
- Aztech CDA268, Orchid CDS-3110 și Okano/Wearnes CDD-110, Conrad TXC, CyCDROM CR520ie, CyCDROM CR540ie, CyCDROM CR940ie
- Creative Labs CD-200(F)
- Funai E2550UA/MK4015
- GoldStar R420

- unitatea CD-ROM IBM ISA externă
- Lasermate CR328A
- LMS Philips CM 206
- Longshine LCS-7260
- unitățile Matsushita/Kotobuki/Panasonic modelele CR-521, CR-522, CR-523, CR-562 și CR-563 (unități le CD-ROM legate împreună cu placa de sunet Sound Blaster Pro)
- unitatea MicroSolutions Backpack, care se conectează la portul paralel al calculatorului
- unități Mitsumi (CR DC LU05S, FX001D/F)
- Optics Storage Dolphin 8000AT
- Sanyo H94A
- Sony CDU31A, CDU33A, CDU-510, CDU-515, CDU-531 și CDU-535
- Teac CD-55A SuperQuad



Sugestie Placa de sunet Sound Blaster 16 oferă la alegere una dintre două interfețe cu unitatea CD-ROM: o interfață proprie sau o interfața SCSI. Sistemul Linux le suportă pe amandouă, dar este necesar să știți de care dintre acestea dispune placa pe care o aveți în configurație. Această informație se găsește de obicei în manualul care însoțește placa Sound Blaster 16.

Observați că puteți conecta o unitate ATAPI CD-ROM la un controller IDE, tratând-o ca pe o a doua unitate de hard-disc. Dacă doriți o modalitate simplă de conectare a unității CD-ROM la PC-ul dumneavoastră și nu aveți nevoie de o unitate suplimentară de hard-disc, puteți lua în calcul varianta instalării unității ATAPI ca slave la unitatea de hard-disc de pe care porniți sistemul.

Placa de sunet

Pe calculatoarele PC, gestiunea plăcilor de sunet și a unităților CD-ROM se efectuează relativ unitar, din cauză că majoritatea aplicațiilor multimedia care utilizează CD-ROM-uri includ efecte sonore de care vă puteți bucura numai dacă dispuneți de o placă de sunet. Și în sistemul Linux, placa de sunet are rolul de a vă permite redarea sunetului. Dacă dispuneți de o astfel de placă, puteți asculta CD-uri audio sau puteți juca Doom (un binecunoscut joc) beneficiind integral de efectele sonore ale acestuia.

Versiunea de Linux de pe CD-ROM-ul atașat suportă următoarele plăci de sunet:

- 6850 UART (Universal Asynchronous Receiver Transmitter) MIDI (Musical Instrument Digital Interface)
- Adlib (OPL2)
- ATI Stereo F/X (compatibilă cu Sound Blaster)
- Audio Excell DSP16
- Aztech Sound Galaxy NX Pro
- plăci bazate pe Crystal CS4232 (PnP – Plug and Play)
- ECHO-PSS (Orchid SW32 și Cardinal DSP16)
- Ensoniq SoundScape (trebuie însă să porniți sistemul DOS pentru a inițializa placa)
- Gravis Ultrasound, Ultrasound MAX și Ultrasound pe 16 biți
- Logitech SoundMan 16 (compatibilă cu PAS-16), SoundMan Games și SoundMan Wave (Jazz16/OPL4)

- MediaTriX AudioTriX Pro
- Media Vision Premium 3D Jazz16 (compatibilă cu Sound Blaster Pro), Pro Audio Spectrum 16 (PAS-16) și Pro Sonic 16 Jazz
- Microsoft Sound System (AD1848)
- MPU-401 MIDI
- plăcile OAK OTI-601D (Mozart)
- plăcile OPTi 82C928/82C929 (MAD16/MAD16 Pro/ISP16/Mozart)
- Sound Blaster, Sound Blaster 16 și Sound Blaster Pro
- Sound Galaxy NX Pro
- Thunder Board (compatibilă Sound Blaster)
- plăcile Turtle Beach Wavefront (Maui, Tropez)
- plăcile WaveBlaster și altele compatibile cu Sound Blaster 16
- plăci bazate pe cipurile ESS Technologies AudioDrive, cum sunt 688, 1688 și 1868

Placa de rețea

Placa de rețea este necesară numai dacă sistemul Linux va fi conectat la o rețea Ethernet. Sistemul Linux suportă o varietate de adaptoare de rețea Ethernet. Adaptoarele Arcnet sau IBM Token Ring sunt de asemenea suportate.



Referință încrucișată

Vezi capitolul 16 pentru mai multe informații despre suportul oferit de Linux pentru adaptoarele Token Ring sau de alte tipuri.

În continuare, vă prezentăm o listă a plăcilor Ethernet pe care sistemul Linux le suportă:

- 3Com 3C503, 3C505, 3C507, 3C509, 3C509B (pentru magistrala ISA) și 3C579 (magistrala EISA)
- plăcile Ethernet 3Com Etherlink III Vortex (3C590, 3C592, 3C595, 3C597) pentru magistrala PCI, 3Com Etherlink XL Boomerang (3C900, 3C905) pentru magistrala PCI și placa 3Com Fast Etherlink (3C515) pentru magistrala ISA
- plăcile Ethernet bazate pe cipul AMD LANCE (79C960) pentru magistralele ISA și PCI
- placa Ethernet Accton pentru portul paralel
- plăcile Allied Telesis AT1500, AT1700 și LA100PCI-T
- Ansel Communications AC3200 pentru magistrala EISA
- Apricot Xen-II
- AT&T GIS WaveLAN
- plăcile Ethernet AT-Lan-Tec și RealTek pentru portul paralel
- Cabletron E21xx (x poate fi orice cifră)
- Cogent EM110
- placa Ethernet D-Link DE600 și DE620 pentru portul paralel
- Danpex EN-9400
- Digital Equipment Corporation (DEC) DEPCA, EtherWORKS, EtherWORKS 3, DE425 (EISA), DE434 (PCI), DE435 (PCI), DE450, DE500, DE450-XA, DE500-XA și DEC QSilver

- Fujitsu FMV-181, FMV-182, FMV-183 și FMV-184
- Hewlett-Packard HP J2405A, PCLAN (27245 și seria 27xxx), PCLAN PLUS (27247B și 27252A) și 10/100VG PCLAN (J2577, J2573, 27248B, J2585) (ISA, EISA și PCI)
- ICL EtherTeam 16i/32 (EISA)
- Intel EtherExpress și EtherExpress Pro
- KTI ET16/P-D2, ET16/P-DC ISA
- New Media Ethernet
- Novell Ethernet NE1000, NE1500, NE2000 și NE2100 (nu se garantează instalarea tuturor clonelor)
- PureData PDUC8028 și PDI8023
- Racal-Interlan NI5210 (bazată pe cipul Ethernet i82586), NI6510 (care utilizează cipul AMD 7990 LANCE și nu funcționează când memoria este mai mare de 16 MB)
- SEEQ 8005
- Schneider & Koch G16
- SMC (Western Digital) WD8003, WD8013, SMC Elite, SMC Elite Plus, SMC Elite 16 Ultra, SMC EtherEZ (ISA), seria SMC 9000, SMC PCI EtherPower 10/100
- Zenith Z-Note și IBM ThinkPad 300
- Znyx 312 Etherarray



Sugestie

Dacă aveți de gând să utilizați sistemul Linux pe un calculator PC izolat de acasă, puteți folosi protocolul Serial Line Internet Protocol (SLIP) sau Point-to-Point Protocol (PPP) pentru a vă conecta la Internet utilizând o conexiune prin dial-up la un furnizor de servicii Internet (Internet Service Provider – ISP). Vezi capitolul 18 pentru mai multe detalii.

Lista de verificare a dispozitivelor fizice

Acum, după ce am văzut un sumar al diferitelor dispozitive fizice suportate de Linux, știți cel puțin în mare dacă aveți o configurație adecvată pentru a putea rula Linux. Dacă sunteți în situația de a cumpara un nou PC pentru a rula Linux, lista dispozitivelor suportate vă va ajuta în stabilirea configurației fizice a noului PC.

În concluzie, parcurgeți lista următoare pentru a vedea dacă sunteți gata de a instala sistemul Linux de pe CD-ROM-ul atașat lucrării:

- Are PC-ul un procesor 80386 sau mai bun decât acesta cu o magistrală ISA, EISA, VLB, MCA sau PCI; cel puțin 8 MB de RAM; o unitate de dischete high-density și o unitate de hard-disc mare (cel puțin 500 MB)? Amintiți-vă, dacă intenționați să rulați atât Windows 95 cât și Linux pe PC-ul dumneavoastră, aveți nevoie de minimum 1 GB de spațiu pe hard-disc.
- Are PC-ul o unitate CD-ROM suportată de Linux? (Aveți nevoie de o astfel de unitate pentru a instala Linux de pe CD-ul atașat acestei lucrări.)



Sugestie

Dacă nu știți ce fel de unitate CD-ROM aveți, puteți fi atent la pornirea sistemului în DOS, urmărind încărcarea driver-ului DOS – acesta afișează un mesaj cu marca unității CD-ROM.

- Vă puteți permite să achiziționați o a doua unitate de hard-disc? (În acest caz, puteți instala Linux pe cea unitate. Instalarea sistemului Linux pe o a doua unitate vă protejează

împotriva distrugerii informației de pe prima, care de regulă are sistemul MS-DOS sau Windows încărcat.)

- Dacă aveți un controller SCSI și un număr de dispozitive SCSI pe care doriți să le utilizați în Linux, este controllerul SCSI suportat de Linux?
- Este placa video suportată de Linux? (Dacă nu, nu veți putea configura și rula sistemul de ferestre X.)
- Este mouse-ul dumneavoastră suportat de Linux? (În caz contrar, nu veți putea configura și rula sistemul de ferestre X.)

După cum și comentariile de după fiecare întrebare o indică, nu este neapărat necesar să răspundeți afirmativ la fiecare dintre întrebări. La primele două, acest răspuns afirmativ este însă obligatoriu, deoarece fără configurația fizică de bază, sistemul dumneavoastră nu va putea rula Linux.



Observație

Dacă doriți să efectuați instalarea pe un al doilea hard-disc, nu mai este necesar să treceți prin etapa de partiționare (divizare) a hard-discului în MS-DOS. Săriți peste următoarele câteva secțiuni și treceți direct la crearea dischetei de pornire (vezi secțiunea „Crearea unei dischete de pornire pentru MS-DOS”). Puteți apoi porni sistemul Linux de pe discheta de pornire și partiționa a doua unitate pentru a o utiliza în Linux.

Partiționarea unității de hard-disc în MS-DOS sau Windows

Dacă PC-ul dumneavoastră are o singură unitate de hard-disc, este foarte probabil că pe acea unitate aveți instalat sistemul Microsoft Windows 95 (sau Windows 3.1). Dacă acea unitate are cel puțin 1 GB, recomandăm să păstrați sistemul Windows instalat chiar dacă veți lucra majoritatea timpului în Linux. Unii din pașii procesului de instalare a sistemului Linux pot fi de altfel efectuați din MS-DOS sau din Windows. Puteți de asemenea avea acces la fișiere Windows din Linux. Veți obține maximul cumulat de avantaje dacă mențineți MS-DOS sau Windows atunci când instalați Linux.



Sugestie

Dacă noul PC are un singur hard-disc mare (de regulă mai mare de 2 GB), dar există două unități – C și D – aceasta înseamnă că discul are deja două partiții. În acest caz, puteți utiliza direct partiția extinsă pentru Linux.

De obicei, discul PC-ului dumneavoastră este configurat ca o singură unitate mare, desemnat prin litera C. Cu excepția situației în care puteți face rost de un al doilea disc sau aveți deja un al doilea disc în acest scop, prima dumneavoastră misiune este de a diviza singurul hard-disc pentru a crea spațiu pentru Linux.

Sarcini în MS-DOS sau Windows înainte de instalarea sistemului Linux

Dacă sunteți începător în MS-DOS sau Windows, câțiva dintre pașii de instalare care trebuie efectuați în MS-DOS s-ar putea să vi se pară dificili. În continuare vă prezentăm câțiva dintre termenii și conceptele pe care trebuie să le cunoașteți pentru a efectua pașii de instalare necesari:

- **Unitatea de pornire (de boot).** Prima unitate de dischete (A) este unitatea de pornire. Dacă introduceți o dischetă în unitatea A și porniți alimentarea calculatorului, PC-ul va încerca în mod automat să pornească de pe dischetă. (Majoritatea noilor PC-uri vă permit să stabiliți dispozitivul de pornire, oricum, în

(continuare)

(continuare)

mod prestabilit, aproape toate PC-urile vin configurate cu prima unitate de dischetă ca dispozitiv de pornire.) Această facilitate ține de construcția calculatorului și nu depinde de sistemul de operare instalat pe hard-discul al acestuia. La majoritatea noilor sisteme, unitatea A este o unitate de dischetă de 3,5 inci. Multe sisteme mai vechi utilizează însă o unitate de dischetă de 5,25 inci ca unitate A. Pentru a instala Linux, aveți nevoie de o unitate de dischetă de pornire high-density.

■ **Partiții.** Un hard-disc fizic poate fi divizat în mai multe părți, fiecare dintre acestea putând fi tratată ca unitate logică de hard-disc de sine stătătoare. Deși majoritatea noilor PC-uri utilizează întreg hard-discul ca pe o singură unitate, puteți partiționa discul în până la patru secțiuni, numite *partiții primare*. Pentru a instala sistemul Linux pe unitatea de hard-disc, trebuie să creați cel puțin două partiții pentru Linux – una pentru sistemul de fișiere Linux și cea de-a doua pentru *spațiul de swap* (memorie virtuală care poate păstra temporar conținutul memoriei fizice). Dacă doriți să mențineți sistemul Windows într-o partiție și de asemenea să instalați Linux, aveți nevoie de trei partiții.

■ **Repartiționarea unui hard-disc.** Dacă discul dumneavoastră are o singură partiție, procesul de creare a mai multor partiții se numește repartitiționare. Pentru a repartitiționa un hard-disc, trebuie să salvați în prealabil conținutul acestuia. Utilizați apoi comanda MS-DOS FDISK pentru a șterge vechea partiție și a crea mai multe partiții noi, restaurând în final vecontrollerelechiul conținut al discului într-una din aceste partiții. Distribuția Red Hat Linux (de pe CD-ROM-ul atașat lucrării) include un program MS-DOS numit *FIPS*, pe care îl puteți utiliza pentru repartitiționarea unui hard-disc fără distrugerea conținutului vechii partiții. Acest utilitar nu este însă garantat să funcționeze perfect, astfel încât rămâne necesară salvarea vecontrollerelelor mai importante fișiere înainte de a încerca reparcontrollerelechiul discului cu FIPS.

■ **Formatarea unui disc.** Programul MS-DOS FDISK doar definește o secțiune a unui hard-disc pentru a

fi utilizată de un sistem de operare cum este MS-DOS. Trebuie însă să pregătiți acea secțiune înainte ca sistemul MS-DOS să o poată utiliza pentru a memora fișiere sau directoare. *Formatarea* este procesul de pregătire a unei partiții pentru memorarea unui sistem de fișiere.

■ **Directoare și fișiere.** În MS-DOS, o unitate este divizată în directoare. Fiecare director, la rândul său, poate conține alte directoare sau fișiere. Fișierul este locul în care se memorează efectiv informația. Directoarele au rolul de a vă ajuta să vă organizați documentele și programele. Toate fișierele utilizate de exemplu de către sistemul de operare Windows se găsesc de obicei în directorul C:\WINDOWS. Directorul C:\WINDOWS\SYSTEM conține câteva fișiere speciale necesare sistemului Windows. După cum observați, caracterul backslash (\) este utilizat ca separator între numele directoarelor. Directorul C:\ este cunoscut sub numele de director rădăcină; WINDOWS este un sub-director al directorului rădăcină, iar SYSTEM este un subdirector al lui WINDOWS. Prin urmare, C:\WINDOWS\SYSTEM este situat cu două niveluri mai jos față de directorul rădăcină.

■ **Numele fișierelor.** Un nume de fișier în MS-DOS constă dintr-un nume de 1 până la 8 caractere, urmat de un punct și de o extensie care poate avea 0 până la 3 caractere. De exemplu, README.TXT este numele unui fișier MS-DOS cu numele README și extensia TXT. Programele executabile au extensia COM sau EXE, în timp ce extensia DOC reprezintă de obicei documentele care pot fi deschise cu ajutorul unui editor de texte.

■ **Comenzi MS-DOS.** Puteți utiliza sistemul MS-DOS prin intermediul comenzilor pe care le introduceți la prompt. Interpretorul de comenzi MS-DOS afișează un prompt (de regulă, numele unității și directorului curent, urmat de semnul „mai mare”, ca de exemplu C:\>). Comenzile MS-DOS au uneori opțiuni care încep cu caracterul slash (/). Puteți utiliza opțiunea /S a comenzii FORMAT, spre exemplu, pentru a formata și copia fișierele sistem pe un disc. Rețineți că puteți utiliza comenzile MS-DOS chiar și în sistemul Windows 95 într-o fereastră MS-DOS.

Pașii operației de repartitiționare a hard-discului

În secțiunile următoare, vom presupune că PC-ul dumneavoastră are un singur hard-disc pe care sistemul MS-DOS este deja instalat (se poate ca și Windows să fie instalat, dar aceasta nu are relevanță pentru operațiile pe care le veți efectua în curând). Pentru a repartitiționa discul, trebuie să executați următoarele operații:

1. Salvați conținutul hard-discului. Dacă tocmai ați cumparat un PC nou, aveți noroc, deoarece discul nu are prea multe date de salvat.
 2. Creați o dischetă de pornire MS-DOS (în Windows 95, creați o dischetă de pornire utilizând opțiunea Add/Remove Programs din Control Panel). Această dischetă o veți utiliza pentru pornirea PC-ului și partiționarea hard-discului.
 3. Rulați programul FDISK și creați noile partiții.
 4. Formatați partiția MS-DOS și restaurați fișierele din copia de siguranță creată la pasul 1.
- Următoarele patru secțiuni vă vor ghida prin acești pași.

Partiționarea discului în Windows 95

Dacă PC-ul dumneavoastră rulează Windows 95, dischetă de pornire utilizând opțiunea Add/Remove Programs din Control Panel și să porniți apoi PC-ul de pe discheta realizată. După aceasta, puteți rula programul FDISK pentru a crea noi partiții.

Salvarea conținutului hard-discului

Salvarea conținutului unui disc este o operație de durată, dar nu există o modalitate sigură de a repartitiționa discul fără a-i face mai întâi o copie de siguranță. Sistemul Red Hat Linux este livrat împreună cu utilitarul FIPS, care poate partiționa un disc fără a distruge datele memorate pe acesta.

Modalitatea de realizare a copiei de siguranță depinde în mod sensibil de informația pe care o aveți memorată pe hard-disc. Dacă aveți un PC nou, discul conține probabil sistemul Windows și alte programe atașate de acesta. Dacă dispuneți de discurile originale cu sistemul Windows și programele atașate, puteți renunța la efectuarea copiei de siguranță, reinstalând totul după crearea noilor partiții. Dacă însă PC-ul dumneavoastră nu este nou, este indicat să salvați numai directoarele pe care nu le puteți reinstala de pe dischetele sau CD-ROM-urile originale. De exemplu, dacă aveți un editor de texte, nu este necesar să salvați și directorul în care acesta este instalat, deoarece îl puteți reinstala. Este suficient să salvați directoarele care conțin documentele pe care le-ați creat cu ajutorul acelui editor de texte.

Pentru a realiza o copie de siguranță a hard-discului, puteți apela la un utilitar livrat împreună cu sistemul MS-DOS. În versiunile DOS 5.0 sau anterioare, utilitarul BACKUP vă permite să salvați întregul conținut al discului sau numai anumite directoare. Pentru versiunile DOS 6.0 și ulterioare, utilitarul este MSBACKUP.

Crearea unei dischete de pornire pentru MS-DOS

De regulă, atunci când alimentați calculatorul, acesta își încarcă automat sistemul de operare (de exemplu, MS-DOS sau Windows 95) de pe prima unitate de hard-disc, unitatea C. Dacă introduceți o dischetă în unitatea A și porniți PC-ul, acesta va încărca sistemul de operare de pe dischetă. Dacă discheta introdusă nu conține sistemul de operare, va apărea următorul mesaj:

Non-System disk or disk error
Replace and strike any key when ready:

Acest mesaj spune că PC-ul a încercat să încarce, dar nu a găsit fișierele sistemului de operare, pe dischetă. Într-o astfel de situație, se recomandă să scoateți discheta și să apăsați o tastă, iar PC-ul va porni de pe unitatea de hard-disc.

Pentru a modifica partițiile hard-discului, este necesar să porniți calculatorul de pe unitatea A. În acest mod, puteți fi siguri că unitatea de hard-disc nu este utilizată în momentul în care îi modificați partițiile. Pentru a crea o dischetă de pornire, executați următorii pași:

1. Introduceți o dischetă în unitatea A și tastați comanda **FORMAT A: /S**. Sistemul MS-DOS vă va răspunde cu următorul mesaj:

Insert new diskette in drive A
and press ENTER when ready...



Conținutul anterior al dischetei se distruge în momentul în care o formatați.

Atenție!

2. Apăsați tasta Enter, deoarece deja ați introdus discheta în unitatea A. Sistemul MS-DOS va formata discheta și va plasa fișierele necesare sistemului de operare pe ea (acesta este rolul opțiunii /S din comanda FORMAT). Sistemul DOS vă va mai cere de asemenea o etichetă de volum (Volume label); puteți apăsa tasta Enter ca răspuns. În fine, sistemul vă întreabă dacă mai doriți să formatați și o altă dischetă; apăsați **N**, indicând terminarea operației de formatare.

3. Copiați celelalte fișiere necesare pe unitatea A.

Va trebui să copiați programul FDISK.EXE pe dischetă pentru a-l putea utiliza la partiționarea hard-discului și programul FORMAT.COM, pentru a formata noua partiție DOS. Utilizați următoarele comenzi pentru copierea programului FDISK.EXE pe discheta din unitatea A (în paranteze, se găsesc comentariile autorului):

```
C:
CD \WINDOWS\COMMAND (comenzile DOS se găsesc în acest director)
COPY FDISK.EXE A:
COPY FORMAT.COM A:
```



Aveți de asemenea nevoie de alte programe pentru a restaura copia de siguranță creată anterior. Nu este necesar ca aceste programe să se găsească pe discheta de pornire; asigurați-vă însă că aveți o copie a lor pe o altă dischetă.

Atenție!

4. Testați discheta de pornire.

Închideți toate aplicațiile aflate în rulare, introduceți discheta nou creată în unitatea A și apăsați Ctrl+Alt+Delete pentru a reporni PC-ul. Sistemul MS-DOS trebuie în acest caz să pornească, afișând promptul A:\>.

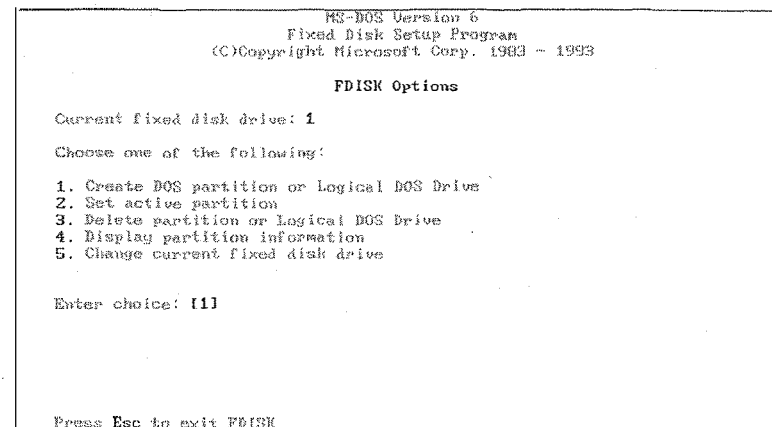
Repartiționarea hard-discului utilizând FDISK

După ce ați realizat copia de siguranță a discului și ați pregătit o dischetă de pornire în MS-DOS, puteți trece la repartiționarea discului. Pentru a începe această operație, introduceți discheta de pornire MS-DOS în unitatea A și reporniți calculatorul (apăsați Ctrl+Alt+Del sau acționați de la butonul de pornire). Când apare promptul A:\>, tastați **FDISK** și apăsați Enter. Programul FDISK se va lansa afișând un mesaj referitor la activarea suportului pentru lucrul cu discuri mari, utilizând un nou sistem de fișiere, numit FAT32. Din cauza faptului

că sistemul de fișiere FAT32 nu este suportat în multe sisteme de operare, inclusiv în versiuni mai vechi de Windows 95, nu se recomandă să activați această facilități a programului FDISK. Apăsați Enter pentru a accepta răspunsul prestabilit, care este No. Programul FDISK va afișa ecranul din figura 1-1.

Figura 1-1

Ecranul inițial în programul FDISK



Atenție!

Salvați conținutul hard-discului înainte de a utiliza FDISK pentru a-l repartiționa (vezi secțiunea precedentă, „Salvarea conținutului hard-discului”). După ce veți modifica partițiile, nu veți mai avea acces la informația memorată anterior pe disc. Puteți să rulați FDISK doar pentru vizualizarea partițiilor curente ale discului, părăsind programul fără a distruge conținutul acestuia, dar se impune o atenție deosebită la introducerea comenzilor, pentru a nu șterge accidental întreg conținutul discului.



Sugestie

FDISK este disponibil în toate versiunile de MS-DOS și în Windows 95. (În Windows 95, puteți porni PC-ul în mod MS-DOS, de unde puteți repartiționa discul utilizând comanda FDISK.) În această sesiune de lucru cu FDISK, ne propunem să ștergem partiția existentă, creând o partiție DOS mai mică, astfel încât să rămână spațiu suficient și pentru partițiile Linux.

Utilizați următoarea strategie pentru a vă repartiționa discul:

1. Selectați opțiunea 4 din meniul FDISK pentru a vizualiza informația referitoare la partiționarea curentă a discului.
 2. Selectați opțiunea 3 din meniu pentru a șterge partiția DOS.
 3. Selectați opțiunea 1 din meniul FDISK pentru a crea o nouă partiție DOS, mai mică, lăsând spațiu suficient și pentru Linux (dimensiunile efective vor fi precizate în secțiunea „Crearea unei noi partiții DOS”).
 4. Selectați opțiunea 2 din meniu pentru a marca noua partiție DOS creată ca fiind activă.
- Ulterior, veți partiționa restul discului în Linux, operație descrisă în secțiunea „Partiționarea și utilizarea hard-discului”.

Verificarea partițiilor existente pe disc

Înainte de a șterge partiția DOS, apăsați 4 și apoi Enter pentru a vizualiza informația referitoare la partiționarea curentă a discului. Figura 1-2 prezintă ecranul FDISK pentru un disc obișnuit care conține o singură partiție DOS pe întreg discul.

Figura 1-2

Ecranul FDISK conținând informația de partiționare standard

```

Display Partition Information

Current fixed disk drive: 1

Partition Status Type Volume Label Mbytes System Usage
C: 1 A PRI DOS LMBPC-C 325 FAT16 100%

Total disk space is 325 Mbytes (1 Mbyte = 1048576 bytes)

Press Esc to continue

```

Discul 1 este primul hard-disc fizic existent în sistem. Valoarea PRI DOS din câmpul Type indică faptul că avem de-a face cu o partiție DOS primară – aceasta conținând fișierele necesare pentru a încărca sistemul MS-DOS. Valoarea din câmpul Partition ne spune că această unitate este unitatea C în MS-DOS. Este foarte probabil ca și dumneavoastră să fiți în această situație.

Ștergerea partiției DOS primare

Apăsați ESC pentru a vă întoarce în meniul principal din FDISK (cel din figura 1-1). Pentru a șterge partiția DOS primară, apăsați 3 și apoi Enter. FDISK va afișa un alt meniu, ilustrat în figura 1-3.

Figura 1-3

Ecranul FDISK cu opțiunile de ștergere a unei partiții

```

Delete DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1. Delete Primary DOS Partition
2. Delete Extended DOS Partition
3. Delete Logical DOS Drive(s) in the Extended DOS Partition
4. Delete Non-DOS Partition

Enter choice: [ 1 ]

Press Esc to return to FDISK Options

```

Puteți șterge mai multe tipuri de partiții. O partiție extinsă, de exemplu, este o partiție care poate fi ulterior subdivizată în mai multe unități logice. De regulă însă, atunci când sistemul Windows ocupă întreg discul, nu aveți de șters decât o singură partiție DOS primară.

Pentru a șterge partiția DOS primară, apăsați 1 și apoi Enter. Programul FDISK va afișa ecranul din figura 1-4, cerând confirmarea operației de ștergere a partiției.

Apăsați Enter. FDISK va șterge partiția și se va întoarce la ecranul principal din figura 1-1.

Crearea unei noi partiții DOS

În acest pas, veți crea din nou partiția DOS primară, de această dată cu o dimensiune mai mică. În acest punct, trebuie să decideți cât spațiu veți păstra pentru MS-DOS (și Windows) și cât veți dedica sistemului Linux. Opțiunea finală depinde de capacitatea totală a discului și de modul în care doriți să utilizați sistemul Linux.

Figura 1-4

Ecranul FDISK cere o confirmare înainte de a șterge partiția DOS primară.

```

Delete Primary DOS Partition

Current fixed disk drive: 1

Partition Status Type Volume Label Mbytes System Usage
C: 1 A PRI DOS LMBPC-C 325 FAT16 100%

Total disk space is 325 Mbytes (1 Mbyte = 1048576 bytes)

WARNING! Data in the deleted Primary DOS Partition will be lost.
What primary partition do you want to delete..? [1]

Press Esc to return to FDISK Options

```

Tabelul următor prezintă o estimare grosieră a spațiului necesar pe disc:

Elemente componente	Spațiul necesar (MB)
Instalare completă Linux de pe CD-ROM-ul atașat	400
Spațiu de swap (utilizat ca memorie virtuală)	16
Spațiu disponibil (pentru a putea lucra în Linux)	150
Windows	500
Total	1 066



Sugestie

Dacă PC-ul dumneavoastră are mai mult de 16 MB de memorie, alocați pentru spațiul de swap aceeași dimensiune cu a memoriei. În caz contrar, rezervați cel puțin 16 MB pe disc pentru spațiul de swap.

Dacă aveți un disc fix de 1,2 GB, alocați 500 MB pentru partiția primară DOS și lăsați restul spațiului pentru Linux. Din nefericire, din cauza numărului mare de aplicații care necesită mult spațiu pe disc, valoarea de 500 MB este în general insuficientă pentru a permite rularea tuturor aplicațiilor Windows. Presupunând însă că doriți în primul rând să utilizați Linux, vă recomandăm să-i rezervați pe disc un spațiu suficient.

Dacă însă dispuneți de un disc cu capacitatea mai mare (majoritatea noilor sisteme sunt livrate cu discuri de peste 2 GB), amintiți-vă aceste valori minime și măriți în mod proporțional dimensiunile partițiilor DOS și Linux.

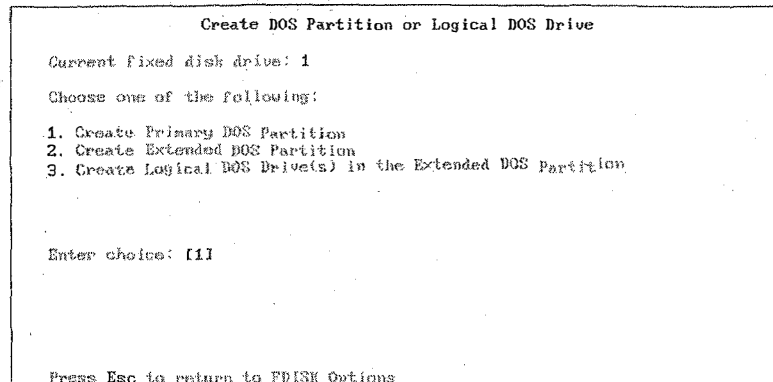
Dacă aveți un calculator PC mai vechi cu un disc de capacitate ceva mai mică – să presupunem 500 MB – este necesar ori să dedicați întregul disc pentru Linux, ori să vă procurați un al doilea disc, astfel încât să puteți menține sistemul Windows și totodată să puteți rula și Linux.

Pentru a crea noua partiție primară DOS, apăsați 1 în ecranul principal FDISK (vezi figura 1-1) și apoi Enter. FDISK va afișa meniul prezentat în figura 1-5.

Apăsați Enter pentru a indica faptul că doriți să creați o partiție DOS primară. Programul FDISK vă va cere să precizați dimensiunea partiției, în megabaiți. Specificați dimensiunea spațiului pe care îl veți rezerva partiției DOS și apăsați Enter. FDISK va crea noua partiție DOS și se va întoarce la ecranul principal din figura 1-1.

Figura 1-5

Ecranul FDISK cu opțiunile de creare a unei noi partiții



Activarea unei partiții DOS

Pentru a putea încărca sistemul MS-DOS de pe o partiție DOS primară, aceasta trebuie mai întâi activată. Din ecranul principal al programului FDISK (vezi figura 1-1), apăsați 2 și apoi Enter. Programul vă va cere numele partiției pe care doriți să o activați. Aceasta este chiar partiția DOS primară pe care ați creat-o. Apăsați Enter pentru a activa această partiție.

Apăsați tasta Esc pentru a părăsi programul FDISK. FDISK vă va informa că va reporni calculatorul. Lăsați discheta de pornire introdusă în unitatea A și apăsați Enter. Calculatorul va reporni, reapărând promptul A:\>.

Restaurarea partiției MS-DOS

Noua partiție DOS primară va deveni unitatea C a PC-ului dumneavoastră. Înainte de a o putea utiliza, trebuie însă să o formatați. Pentru a formata unitatea C, tastați **FORMAT C: /S**. Programul FORMAT afișează un mesaj de avertizare. În acest caz, puteți ignora mesajul, deoarece formatați o partiție nou creată pe hard-disc.

După ce ați formatat unitatea C, este momentul să restaurați conținutul hard-discului. Dacă ați salvat numai fișierele proprii și nu și fișierele sistemului Windows, trebuie să începeți prin instalarea acestuia pe noua unitate C. De obicei, este suficient să porniți PC-ul de pe prima dischetă de instalare a sistemului Windows și să urmați mai departe instrucțiunile.

După ce ați reînaltat sistemul Windows pe unitatea C, puteți utiliza programul de restaurare pentru a reface toate fișierele salvate anterior de pe disc. Pentru a realiza această operație, trebuie să urmați întocmai instrucțiunile programului de salvare pe care l-ați utilizat. Dacă spre exemplu ați utilizat programul MSBACKUP, disponibil în versiunea DOS 6.0 și în versiunile ulterioare, veți utiliza același program pentru a reface fișierele de pe dischetele pe care au fost salvate.

Repartiționarea utilizând FIPS

Dacă priviți partițiile ca porțiuni ale hard-discului, vă veți întreba dacă există un mod de a strânge laolaltă partea neutilizată a discului, creând o nouă partiție din aceasta fără a distruge informația existentă pe disc. Aceasta este ideea care stă la baza unui program numit *FIPS* (The First Nondestructive Interactive Partition Splitting Program – primul program interactiv de partiționare nedestructivă). FIPS poate despărți orice partiție primară DOS existentă în două partiții.



Sugestie

Deși nu aveți nici o garanție că FIPS va reuși să despartă cu succes o partiție existentă, îl puteți utiliza pentru a crea spațiu pentru sistemul Linux, în special atunci când PC-ul este nou, având doar sistemele DOS și Windows instalate pe disc. În acest caz, chiar dacă FIPS nu funcționează corect, puteți reinstala Windows și aplicațiile. Am utilizat FIPS pentru a despărți partiția DOS de pe discul unui PC nou, cu scopul de a crea spațiu pentru sistemul Linux.

Programul FIPS.EXE și fișierele atașate acestuia se găsesc în subdirectorul \DOSUTIL de pe CD-ROM-ul atașat. Pentru a utiliza programul FIPS, urmați pașii:

1. Pentru ca FIPS să funcționeze corect, toate zonele utilizate de pe disc trebuie să fie contigue sau pe cât posibil foarte bine compactate. Puteți pregăti discul pentru FIPS rulând mai întâi un program de defragmentare. În sistemul MS-DOS 6.0 sau într-o versiune ulterioară acestuia, utilizați programul DEFRAG pentru a defragmenta discul. În Windows 95, executați clic-dreapta pe pictograma care reprezintă discul în Explorer, selectați Properties din meniul care apare, executați apoi clic pe eticheta Properties și aici pe butonul Defragment Now. Este de asemenea recomandat să verificați dacă discul nu are erori, utilizând un program cum este Norton Disk Doctor (în Windows 95, utilizați SCANDISK). Dacă din întâmplare aveți programul Norton Speed Disk, îl puteți utiliza pentru a defragmenta discul, deoarece este sensibil mai rapid decât utilitarul DEFRAG.

2. Creați o dischetă de pornire, utilizând comanda **FORMAT A: /S**.

3. Copiați următoarele fișiere de pe CD-ROM pe discheta nou formatată (în continuare, se presupune că D: este unitatea CD-ROM):

```
COPY D:\DOSUTILS\FIPS.EXE A:
COPY D:\DOSUTILS\RESTORRB.EXE A:
COPY D:\DOSUTILS\FIPSDOCS\ERRORS.TXT A:
```

FIPS.EXE este programul care realizează despărțirea partițiilor. ERRORS.TXT reprezintă o listă a mesajelor de eroare afișate de FIPS. Puteți consulta această listă pentru a obține o explicație a oricărui mesaj de eroare pe care FIPS îl afișează. RESTORRB.EXE este programul care vă permite să restaurați anumite părți importante ale discului dintr-o copie a acelor zone, creată de FIPS.

4. Lăsați discheta de pornire în unitatea A și apăsați Ctrl+Alt+Delete pentru a reporni PC-ul. Acesta va porni de pe unitatea A, afișând promptul A:\>.

5. Tastați **FIPS**. Se va lansa programul FIPS, acesta prezentându-vă informații despre discul dumneavoastră. Programul FIPS vă oferă posibilitatea de a efectua o copie de siguranță a unor zone importante de pe disc înainte de a începe repartiționarea. După aceasta, FIPS indică primul cilindru liber din care poate începe noua partiție (și dimensiunea acesteia, în megabaiți). Figura 1-6 prezintă rezultatul afișat de programul FIPS în această etapă.

6. Utilizați săgețile spre stânga și dreapta pentru a stabili cilindrul cu care va începe noua partiție (cea care va rezulta din despărțirea celei existente), modificând astfel dimensiunea noii partiții. Apăsând săgeata spre dreapta, numărul cilindrului de pornire va crește (astfel, se va lăsa mai mult spațiu în partiția deja existentă, reducându-se dimensiunea noii partiții create).

7. Atunci când sunteți mulțumit de dimensiunea noii partiții, apăsați Enter. FIPS va afișa tabela de partiții modificată și vă va cere să apăsați fie C pentru a continua, fie R (Reedit), pentru a modifica încă o dată tabela de partiții.

8. Apăsați C pentru a continua. FIPS va afișa o informație care se referă la disc și vă va întreba dacă doriți să scrieți noua tabelă de partiții pe disc.

9. Apăsați **Y**. FIPS va scrie noua tabelă de partiții pe disc și apoi se va închide.

Figura 1-6

Programul FIPS cere utilizatorului dimensiunea noii partiții

```
Sectors per FAT: 256
Sectors per track: 63
Drive heads: 255
Hidden sectors: 63
Number of sectors (long): 4192982
Physical drive number: 80h
Signature: 29h

Checking boot sector ... OK
Checking FAT ... OK
Searching for free space ... OK

Do you want to make a backup copy of your root and boot sector before
proceeding (y/n)? y
Do you have a bootable floppy disk in drive A: as described in the
documentation (y/n)? y

Writing file a:\rootboot.000

Enter start cylinder for new partition (17 - 260):

Use the cursor keys to choose the cylinder, <enter> to continue

Old partition      Cylinder      New Partition
1059.0 MB          135          988.4 MB
```

Scoateți discheta din unitatea A și reporniți PC-ul. Când sistemul va porni din nou, toată informația de pe disc trebuie să fie intactă, dar unitatea C va fi mai mică. Am creat astfel o partiție din părțile neutilizate ale unității C inițiale.

Nu mai este nevoie de nici o operație asupra partiției nou create în sistemul DOS. Mai târziu, în secțiunea „Partiționarea și utilizarea hard-discului”, veți învăța cum să folosiți noua partiție în Linux.

Crearea dischetei de pornire a sistemului Red Hat

După ce ați repartiționat hard-discul, creând spațiu pentru Linux, puteți trece la următorul pas al instalării sistemului Linux de pe CD-ROM-ul atașat acestei lucrări: crearea dischetei de pornire Linux. (În acest pas, veți porni mai întâi calculatorul fără nici o dischetă în unitatea A, rulând apoi ca de obicei Windows.)



Sugestie

Nu aveți neapărat nevoie de o dischetă dacă puteți porni PC-ul în MS-DOS – nu într-o fereastră DOS din Windows 95 – explorând CD-ROM-ul de la promptul de comandă DOS. Dacă rulați Windows 95, este suficient să reporniți calculatorul în modul MS-DOS din cauza fișierelor de pornire – AUTOEXEC.BAT și CONFIG.SYS – care s-ar putea să nu fie corect configurate. Încercați să reporniți calculatorul în modul MS-DOS și cercetați dacă puteți avea acces la CD-ROM. Dacă reușiți, săriți peste această secțiune și treceți direct la secțiunea „Pornirea sistemului Linux pentru instalare”.

Ca și discheta de pornire pentru MS-DOS, discheta de pornire Linux este utilizată pentru a porni calculatorul dumneavoastră sub Linux. După ce ați instalat sistemul Linux, nu mai aveți nevoie de dischetă cu excepția cazurilor de urgență (când trebuie să reinstalați sistemul de pe CD-ROM).

Discheta de pornire conține o versiune inițială a sistemului Linux utilizată pentru pornirea sistemului, pregătirea hard-discului și încărcarea restului sistemului de operare. Crearea dischetei de pornire se realizează cu ajutorul utilitarului RAWRITE.EXE, care copiază un fișier special, numit *image de pornire a sistemului Linux*, pe o dischetă.

Pentru a crea o dischetă de pornire pentru Linux din Windows, executați următoarele operații:

1. Deschideți o fereastră MS-DOS (selectați MS-DOS Prompt din zona Programs din meniul Start).
2. În fereastra MS-DOS, introduceți următoarele comenzi la promptul MS-DOS (comentariile autorului sunt scrise între paranteze, iar datele pe care trebuie să le introduceți dumneavoastră sunt scrise îngroșat):

```
D: (utilizați litera care reprezintă unitatea CD-ROM)
CD \DOSUTILS
RAWRITE
Enter disk image source file name: \images\boot.img
Enter target diskette drive: A
Please insert a formatted diskette into drive A: and press -ENTER-:
```

După cum vi se sugerează, introduceți o dischetă formatată în unitatea A a PC-ului și apoi apăsați tasta **Enter**. Utilitarul RAWRITE va copia fișierul imagine de pornire pe dischetă.

După ce promptul DOS reappare, puteți scoate discheta din unitatea A și (dacă nu ați făcut deja aceasta) să o etichetați corespunzător. O etichetă de genul *Dischetă de pornire pentru Linux* ni se pare adecvată.

Crearea dischetei suplimentare de instalare pentru Red Hat

Dacă aveți dispozitive care utilizează interfața PCMCIA (de regulă prezentă la PC-urile notebook), mai aveți nevoie de încă o dischetă pe lângă cea de pornire. A doua dischetă poartă numele de *dischetă suplimentară de instalare* și dispune de un sistem de fișiere Linux cu un număr de directoare. Discheta conține programe și fișiere necesare în împrejurări speciale, cum ar fi în cazul instalării de pe o unitate CD-ROM PCMCIA.



Sugestie

Este bine să creați întotdeauna discheta suplimentară de instalare, deoarece aceasta este de asemenea necesară pentru rezolvarea oricăror probleme care pot apărea cu sistemul Linux după ce terminați instalarea.

Pentru crearea dischetei suplimentare, introduceți CD-ROM-ul cu Red Hat Linux în unitatea CD-ROM și tastați următoarele comenzi într-o fereastră MS-DOS:

```
d: (utilizați litera care reprezintă unitatea CD-ROM)
cd \images
\dosutils\rawrite
Enter disk image source file name: supp.img
Enter target diskette drive: a
Please insert a formatted diskette into drive A: and
press -ENTER-:
```

Introduceți o dischetă formatată în unitatea A și apăsați tasta **Enter**. Când promptul DOS reappare, discheta suplimentară este gata.

Pornirea sistemului Linux pentru instalare

Programul de instalare a distribuției Red Hat Linux rulează sub Linux – ceea ce înseamnă că trebuie să rulați deja o versiune de Linux pe PC înainte de a trece la pașii de instalare. Această versiune inițială de Linux poate fi memorată pe o dischetă de pornire sau pe CD-ROM-ul înșuși. Versiunea inițială are rolul de a rula programul de instalare, care pregătește partițiile discului și copiază toate fișierele necesare de pe CD-ROM pe disc.

Instalarea sistemului Linux din rețea utilizând NFS

Puteți instala sistemul Linux de pe un alt sistem, dacă ambele sisteme sunt într-o aceeași rețea, iar celălalt sistem dispune de distribuția Red Hat Linux într-un director care este exportat prin intermediul protocolului NFS (Network File System). Pentru a efectua o astfel de instalare, trebuie să aveți cunoștințe despre ceea ce reprezintă NFS sau să cereți ajutorul administratorului rețelei. Trebuie și în acest caz să creați discheta de pornire ca de obicei. După ce utilizați discheta pentru a porni sistemul,

precizați NFS ca metodă de instalare, oferind informații despre placa de rețea și despre rețea, cum ar fi adresele IP pentru PC-ul dumneavoastră, pentru gateway-ul utilizat și pentru serverul de nume (Name Server). În continuare, este necesar să precizați numele serverului NFS sau adresa IP a acestuia, precum și numele directorului în care se găsește distribuția Red Hat Linux. Puteți apoi continua cu pașii obișnuiți de instalare.

Puteți porni PC-ul cu o versiune inițială a sistemului de operare Linux în oricare din următoarele două moduri:

- Încărcați sistemul Linux executând fișierul de comenzi AUTOBOOT.BAT în timp ce PC-ul rulează MS-DOS.
- Porniți PC-ul de pe discheta pe care ați creat-o ceva mai devreme.

Secțiunile următoare prezintă cele două modalități de pornire a sistemului Linux și operația de inițiere a instalării distribuției Red Hat.

Pornirea sistemului Linux de pe CD-ROM-ul Red Hat

Puteți porni sistemul Linux direct de pe CD-ROM în timp ce PC-ul rulează MS-DOS. Un program MS-DOS, numit LOADLIN.EXE, poate încărca un nucleu (kernel) al sistemului Linux în memorie, lansând totodată sistemul Linux. Nucleul sistemului Linux se găsește într-un alt fișier. Nu este necesar să înțelegeți de acum toate detaliile mecanismului prin care LOADLIN pornește sistemul Linux. De fapt, CD-ROM-ul Red Hat conține un fișier de comenzi DOS – AUTOBOOT.BAT – în directorul \DOSUTILS, care lansează programul LOADLIN cu argumentele corespunzătoare.



Observație

Puteți utiliza AUTOBOOT pentru a lansa direct de pe CD-ROM sistemul Linux numai dacă PC-ul dumneavoastră rulează doar MS-DOS (și nu într-o fereastră MS-DOS din Windows 95). De asemenea, trebuie să puteți avea acces la CD-ROM din MS-DOS. Dacă PC-ul dumneavoastră rulează Windows 95, selectați Shutdown din meniul Start și executați clic pe butonul etichetat Restart the Computer in MS-DOS mode. Dacă de la promptul DOS puteți vizualiza directorul unității CD-ROM (cu comanda DIR D:, unde D este litera prin care este reprezentată unitatea CD-ROM), veți putea ulterior să porniți sistemul Linux direct de pe CD-ROM. În caz contrar, este necesar să utilizați o dischetă de pornire.

Pentru a porni sistemul Linux, introduceți CD-ROM-ul Red Hat în unitatea CD-ROM și introduceți următoarele comenzi de la promptul DOS:

```
D:
cd \dosutils
autoboot
```

După ce sistemul Linux pornește, începe să ruleze programul de instalare a distribuției Red Hat. Secțiunea „Instalarea sistemului Linux de pe CD-ROM-ul Red Hat” descrie în detaliu pașii instalării.

Pornirea de pe discheta Linux

Pentru a porni sistemul Linux în vederea instalării, introduceți discheta de pornire în unitatea A a PC-ului și reporniți PC-ul (apăsând butonul „Reset” sau combinația de taste Ctrl+Alt+Delete). PC-ul dumneavoastră parcurge secvența obișnuită de pornire, cum ar fi verificarea memoriei și rularea codului din ROM BIOS. PC-ul încarcă apoi sistemul Linux de pe dischetă și începe rularea programului de instalare a distribuției Red Hat.

Supravegherea operației de pornire

La câteva momente de la începerea operației de pornire, va apărea un ecran inițial – pe ecran, apare un mesaj de „bine ați venit !” și apoi promptul boot:. Mesajul vă informează că puteți solicita help apăsând una dintre tastele funcționale – de la F1 până la F4.

Dacă doriți să citiți ecranele de help, apăsați tasta funcțională corespunzătoare informației pe care o solicitați. Dacă nu răspundeți pentru mai mult de un minut, procesul de pornire continuă cu încărcarea nucleului Linux în memoria PC-ului. Pentru a începe imediat încărcarea sistemului Linux, apăsați tasta Enter. După ce nucleul Linux este încărcat, acesta pornește în mod automat programul de instalare a distribuției Red Hat.



Sugestie

Pe măsură ce nucleul Linux este încărcat, pe ecran apar diferite mesaje. Aceste mesaje vă informează dacă nucleul a detectat componentele hardware din configurația calculatorului dumneavoastră. De obicei, aceste mesaje defilează mult prea repede pentru a putea fi urmărite. După aceasta, pe ecran apare un mesaj în care sunteți întrebat dacă dispuneți de un monitor color. În acest punct, puteți apăsa Shift+PgUp pentru a defila înapoi și a citi mesajele referitoare la componentele hardware. Aici trebuie să întâlniți mesaje corespunzătoare componentelor hardware ale PC-ului dumneavoastră. În mod special, căutați un mesaj referitor la CD-ROM, deoarece nucleul trebuie mai întâi să detecteze CD-ROM-ul înainte de a putea continua cu restul instalării.

Puteți de asemenea să vizualizați mesajele apăsând combinația Alt+F4 – va apărea un alt ecran virtual, pe care apar mesajele nucleului.

Listingul următor prezintă secvența obișnuită de mesaje afișate la pornirea unui PC cu ajutorul dischetei de pornire Red Hat (în cazul de mai sus, este vorba de un PC cu procesor Pentium cu frecvența de 200 MHz, cu 64 MB de memorie RAM, disc fix de 4 GB, unitate CD-ROM ATAPI și unitate de bandă Iomega ZIP drive):

```
Boot:
Loading initrd.img.....
Loading vmlinuz.....
Uncompressing Linux...done.
Now booting kernel
Console: 16 point font, 400 scans
Console: colour VGA+ 80X25, 1 virtual console (max 63)
pcibios_init: BIOS32 Service Directory structure at 0X000f69f0
pcibios_init: BIOS32 Service Directory entry at 0Xfd7e0
pcibios_init: PCI BIOS revision 2.10 entry at 0Xfd9df
Probing PCI hardware
Calibrating delay loop... ok - 299.01 BogoMIPS
Memory: 62188/65536K available (736 kernel code, 384 reserved, 1256k data)
Swansea University Computer Society NET3.035 for Linux 2.0
NET3: Unix domain sockets 0.13 for Linux NET3.035.
Swansea University Computer Society TCP/IP for NET3.034
IP Protocols: IGMP, ICMP, UDP, TCP
VFS: Diskquotas version dquot_5.6.0 initialized
```

```

Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Checking 'hlt' instruction... Ok.
Linux version 2.0.34 (root@porky.redhat.com) (gcc version 2.7.2.3) #1
Fri May 8 16:05:57 EDT
Starting kswapd v 1.4.2.2
Serial driver version 4.1.3 with no serial options enabled
tty00 at 0X3f8 (irq = 4) is a 16550A
PS/2 auxiliary pointing device detected -- driver installed.
Real Time Clock Driver v1.07
Ramdisk driver initialized : 16 ramdisks of 4096K size
ide: i82371 PIIX (Triton) on PCI bus 0 function 57
    ide0: BM-DMA at 0Xfcf0-0Xfcf7
    ide1: BM-DMA at 0Xfcf8-0Xfcff
hda: ST34342A, 4130MB w/Okb Cache, CHS=523/255/63
hdc: Pioneer CD-ROM ATAPI Model DR-A24X 0105, ATAPI CDRom drive
hdd: IOMEGA ZIP 100 ATAPI, ATAPI FLOPPY drive
ide0 at 0X1f0-0X1f7,0X3f6 on irq 14
ide1 at 0X170-0X177,0X376 on irq 15
Floppy drive(s): fd0 is 1.44M
FDC 0 is a National Semiconductor PC87306
md driver 0.35 MAX_MD_DEV=4, MAX_REAL=8
scsi: 0 hosts.
scsi: detected total.
Partition check:
hda: hda1 hda2 hda3
ide-floppy: hdd: I/O error, pc = 5a, key = 5, asc = 24, ascq = 0
ide-floppy: Can't get drive capabilities
hdd: 98304kB, 96/64/32 CHS, 4096 kBps, 512 sector size, 2941 rpm
hdd: The drive reports both 100663296 and 0 bytes as its capacity
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem)
Welcome to Red Hat Linux

```

În acest exemplu, mesajele arată că sistemul Linux a detectat unitatea CD-ROM ATAPI. Literele hdc se referă la numele de dispozitiv cu care sistemul Linux va denumi unitatea CD-ROM.

Instalarea sistemului Linux de pe CD-ROM-ul Red Hat

După ce porniți versiunea inițială de Linux utilizând una dintre metodele descrise în secțiunea „Pornirea sistemului Linux pentru instalare”, această versiune va rula programul de instalare al sistemului Red Hat Linux de pe CD-ROM. Restul procesului de instalare are loc sub controlul acestui program de instalare.

Interacțiunea cu programul de instalare Red Hat

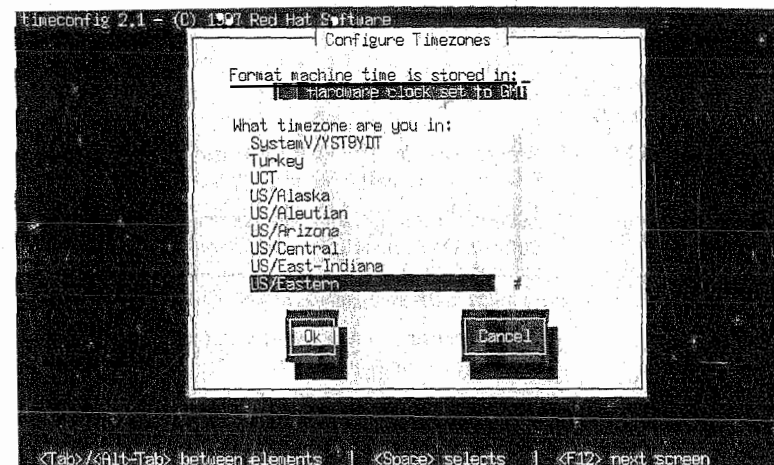
Programul de instalare utilizează o interfață în mod text. În figura 1-7 puteți vedea cum arată un ecran obișnuit de instalare pentru sistemul Red Hat Linux.

Fiecare ecran prezintă de obicei câte o casetă de dialog cu diferite elemente cum ar fi liste de elemente din care veți selecta anumite elemente sau butoane pentru a indica o anumită acțiune. De obicei, aceste butoane sunt etichetate cu OK și Cancel. În josul ecranului, este afișat un mesaj de help, care vă explică cum puteți naviga prin fereastra în mod text.

În locul mouse-ului, veți utiliza anumite taste pentru deplasarea cursorului text. Consultați tabelul 1-1 pentru a afla secvențele de taste asociate diferitelor acțiuni.

Figura 1.7

Un ecran obișnuit de instalare cu o casetă de dialog



Tabelul 1-1 Interacțiunea cu programul de instalare al sistemului Red Hat Linux

Taste	Acțiuni
Săgeți	Deplasare de la un element la altul în caseta de dialog.
Tab	Deplasare la următorul element din caseta de dialog.
Alt+Tab	Deplasare la elementul precedent din caseta de dialog.
Space	Selectează elementul pe care este poziționat cursorul. Dacă elementul este deja selectat, anulează selecția. Prin urmare, tasta Space comută starea de selecție.
Enter	Apasă butonul pe care este poziționat cursorul. Dacă în caseta de dialog se găsește un singur buton, apăsarea tastei Enter este echivalentă cu apăsarea aceluia buton.
F12	Acceptă valorile curente și trece la următoarea casetă de dialog. Este echivalentă cu apăsarea butonului OK.

Monitorizarea procesului de instalare

Pe parcursul instalării, ceea ce trebuie să faceți în primul rând este să răspundeți la diferite casete de dialog, introducând informații necesare programului de instalare. Programul de instalare afișează informații utile pe un număr de console virtuale – acestea reprezintă ecrane de text în memorie, care pot fi vizualizate pe ecranul fizic apăsând secvențele de taste prezentate în tabelul 1-2.



Sugestie

Veți lucra majoritatea timpului pe consola principală – consola virtuală 1. Pentru a trece la o altă consolă virtuală, apăsați secvența de taste prezentată în tabelul 1-2. De exemplu, pentru vizualizarea raportului asupra instalării pe consola virtuală 3, apăsați Alt+F3. După vizualizarea raportului, apăsați Alt+F1 pentru a vă întoarce pe consola principală, astfel încât să puteți continua procesul de instalare.

De obicei, vă puteți descurca fără să fie necesar să treceți pe alte console, dar, dacă apare ceva în neregulă, puteți comuta la ecranul care afișează starea instalării apăsând Alt+F3 pentru a obține mai multe detalii asupra problemei apărute.

Tabelul 1-2 Consolele virtuale pe parcursul procesului de instalare a sistemului Red Hat Linux

Consola virtuală	Secvența de taste	Descriere
1	Alt+F1	Aceasta este consola principală, în care programul de instalare afișează interfața în mod text cu ajutorul căreia se instalează sistemul Linux.
2	Alt+F2	Această consolă afișează promptul unui shell la care puteți introduce comenzi Linux, pentru a monitoriza progresul operației de instalare. Promptul apare numai după ce introduceți CD-ROM-ul și apăsați Enter ca răspuns la caseta de dialog afișată de programul de instalare.
3	Alt+F3	Această consolă conține raportul asupra instalării. Mesajele trimise de programul de instalare apar aici.
4	Alt+F4	Nucleul sistemului Linux afișează mesajele proprii pe această consolă. După pornirea inițială a sistemului Linux, este util să comutați pe această consolă pentru a vizualiza mesajele nucleului, deoarece acestea oferă informații despre dispozitivele fizice detectate de Linux în PC-ul dumneavoastră.
5	Alt+F5	Această consolă prezintă ieșirea tuturor programelor rulate pe parcursul procesului de instalare.

Semnificația fazelor procesului de instalare a sistemului Red Hat

Instalarea sistemului Linux este un proces de durată, care trece prin următoarele etape majore:

- 1. Pregătirea instalării:** Indicați tipul tastaturii și precizați unde se află distribuția Red Hat Linux (de exemplu, pe CD-ROM). Dacă PC-ul dumneavoastră are un dispozitiv PC Card cu interfață PCMCIA, această fază îl detectează și cere introducerea dischetei suplimentare de instalare, care conține fișierele necesare pentru suportul dispozitivelor PCMCIA.
- 2. Partiționarea și utilizarea hard-discului:** În această fază, veți răspunde la întrebări referitoare la dispozitivele SCSI prezente în sistem și veți efectua partiționarea discului. Veți mai preciza de asemenea partiția care va fi utilizată ca spațiu de swap, precum și partiția care va conține directorul rădăcină (reprezentat prin caracterul /). Partițiile sunt de asemenea formate înainte de a intra în uz. Sistemul Red Hat include programul Disk Druid, care execută toți acești pași în această fază a instalării.
- 3. Selectarea componentelor ce vor fi instalate:** Selectați care dintre componentele sistemului Red Hat Linux – cum ar fi sistemul X Window, editorul Emacs sau serverul de Web – doriți să le instalați. După ce selectați componentele, programul de instalare preia controlul și instalează componentele selectate pe disc.
- 4. Configurarea sistemului Linux:** În această etapă, veți configura sistemul X Window, rețeaua TCP/IP, fusul orar precum și orice imprimantă. Veți stabili de asemenea o parolă pentru root – superutilizatorul din sistem. Această etapă și totodată și procesul de instalare se încheie prin precizarea locului în care va fi memorat programul Linux Loader (LILO).

Dacă dispuneți de toate informațiile de configurare (cum ar fi adresele IP și numele calculatorului pentru configurarea rețelei TCP/IP) și totul decurge normal, instalarea sistemului Linux de pe CD-ROM-ul atașat lucrării pe un calculator PC Pentium rapid (cu viteza procesorului de cel puțin

166 MHz) trebuie să dureze aproximativ o oră. De exemplu, pe un PC Pentium cu 200 MHz cu 64 MB de memorie RAM și o partiție de 1 GB alocată pentru Linux, întreaga instalare a durat circa 40 de minute. Pe PC-urile 486, instalarea va dura ceva mai mult.



Observație

Programul de instalare a sistemului Red Hat cercetează – adică încearcă să detecteze prezența – dispozitivelor fizice specifice și modifică în mod corespunzător pașii de instalare. De exemplu, dacă programul detectează prezența unei plăci PC Card, care utilizează interfața PCMCIA, cere automat discheta suplimentară de instalare. Prin urmare, este posibil să apară anumite diferențe în ceea ce privește secvența de pași parcursă la instalare, în funcție de configurația hardware specifică.

Secțiunile următoare descriu detaliat fiecare din fazele de instalare a sistemului Red Hat Linux.

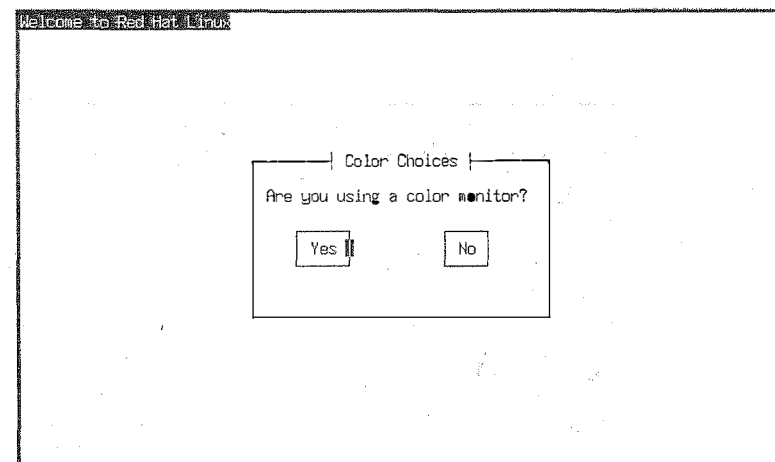
Pregătirea instalării

În această etapă, veți parcurge următorii pași înainte de a începe configurarea discului și instalarea efectivă a sistemului Linux:

- 1. Programul de instalare afișează o casetă de dialog alb-negru pentru a vă întreba dacă dispuneți de un monitor color (vezi figura 1-8). Apăsați tasta Tab pentru a vă deplasa între opțiunile Yes și No. Când cursorul ajunge pe răspunsul corect, apăsați Enter.**

Figura 1-8

Casetă de dialog vă cere să precizați dacă aveți un monitor color.



- 2. În acest moment, apare un ecran introductiv. Mesajul afișat vă invită să vizitați situl Web al firmei Red Hat la adresa:**

<http://www.redhat.com>

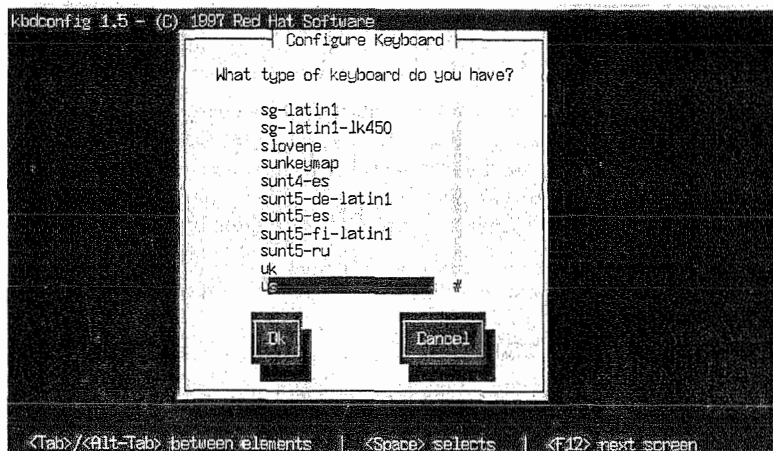
Apăsați Enter pentru a continua instalarea.

- 3. Programul de instalare verifică existența unor dispozitive PC Card care utilizează interfața PCMCIA. Dacă se detectează un astfel de dispozitiv, programul de instalare afișează o casetă de dialog care vă cere să introduceți discheta suplimentară de instalare Red Hat. Dacă aveți un dispozitiv PCMCIA (cum ar fi o placă Ethernet pe un calculator portabil și doriți o instalare prin rețea), introduceți discheta suplimentară de instalare și apoi selectați OK.**

- 4. Programul de instalare afișează o listă cu tipuri de tastaturi, după cum se poate observa și în figura 1-9.**

Figura 1-9

Selectarea tipului tastaturii la instalarea sistemului Red Hat

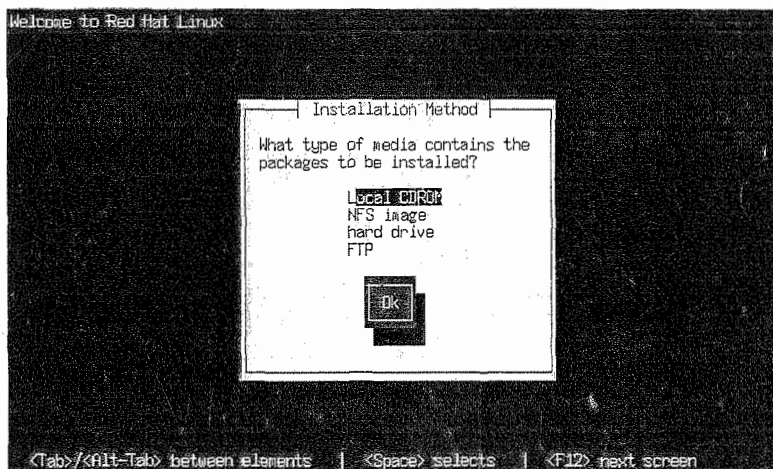


Utilizați tastele direcționale pentru deplasarea în sus sau în jos prin listă și apăsați Spacebar. Selectați tipul de tastatură în conformitate cu cea prezentă în sistemul dumneavoastră. Apăsați apoi tasta F12 pentru a trece la pasul următor.

5. Precizați unde se găsește distribuția Red Hat în caseta de dialog prezentată în figura 1-10.

Figura 1-10

Precizarea locului în care se găsește distribuția Red Hat Linux.



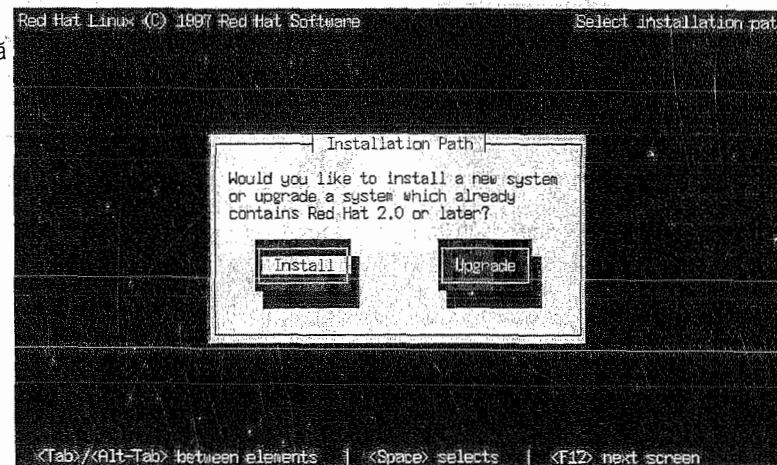
Opțiunea Local CDROM se referă la unitatea CD-ROM de pe PC-ul dumneavoastră. Cum efectuați instalarea de pe CD-ROM-ul atașat acestei lucrări, trebuie să introduceți CD-ROM-ul în unitate, să selectați opțiunea Local CDROM și apoi să apăsați Enter.

Puteți utiliza opțiunile NFS sau FTP pentru a instala sistemul Red Hat Linux de pe alt sistem din rețea. Opțiunea „hard drive” se referă la modalitatea de instalare a sistemului Red Hat Linux cu ajutorul fișierelor copiate pe o altă partiție a hard-discului, utilizând un alt sistem de operare, cum ar fi Windows 95.

6. Programul de instalare inițializează CD-ROM-ul și afișează o casetă de dialog (vezi figura 1-11) în care vi se cere să precizați dacă doriți instalarea unui nou sistem sau actualizarea unei instalări mai vechi a sistemului Red Hat. Presupunând că efectuați pentru prima dată instalarea, asigurați-vă de prezența cursorului pe butonul Install și apăsați Enter.

Figura 1-11

Din această casetă de dialog puteți selecta o nouă instalare sau o actualizare.



7. Programul de instalare afișează apoi ecranul prezentat în figura 1-12, care vă întreabă dacă PC-ul dispune de un adaptor SCSI.

Dacă aveți orice dispozitiv SCSI, cum ar fi o unitate de disc SCSI, trebuie să selectați Yes. Programul de instalare va afișa apoi o listă de adaptoare SCSI (vezi figura 1-13) din care trebuie să-l selectați pe cel corespunzător sistemului dumneavoastră.

Programul de instalare încarcă apoi driverul pentru adaptorul SCSI. După acest pas, puteți instala sistemul Linux pe unitatea de disc SCSI din configurația dumneavoastră.

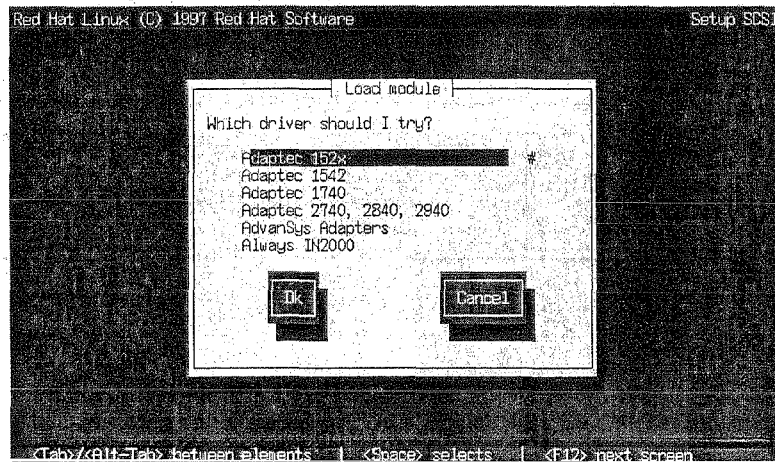
Figura 1-12

Selectați Yes dacă în PC-ul dumneavoastră se găsește un adaptor SCSI.



Figura 1-13

Selectarea unui driver pentru eventualul adaptor SCSI.



Următoarea etapă majoră a instalării implică partiționarea hard-discului pentru a putea fi utilizat în sistemul Linux.

Partiționarea și utilizarea hard-discului

Ca și MS-DOS, sistemul Linux necesită partiționarea și pregătirea în prealabil a hard-discului pe care va fi instalat. De obicei, nu veți efectua acest pas în MS-DOS, deoarece atunci când cumpărați calculatorul de la o firmă, acea firmă se ocupă de partiționarea discului și de instalarea sistemului Windows și a tuturor aplicațiilor specifice pe disc. Cum însă acum efectuați o instalare de la zero a sistemului Linux, este necesar să efectuați chiar dumneavoastră acest pas crucial. Este necesar să pregătiți partițiile discului sub Linux înainte de a putea instala efectiv sistemul de operare. După cum veți vedea în secțiunile următoare, această activitate se reduce la urmarea unor instrucțiuni clare.

Atunci când programul de instalare a sistemului Red Hat Linux ajunge în faza de partiționare a discului, afișează o casetă de dialog în mod text cu două opțiuni pentru partiționarea și utilizarea hard-discului:

■ **Disk Druid:** Acesta este un utilitar Red Hat care utilizează casete de dialog în mod text pentru a vă permite să partiționați discul specificând în același timp ce părți ale sistemului de fișiere Linux vor fi încărcate în fiecare partiție.

■ **fdisk:** Acesta este programul standard de partiționare din Linux, similar în concepție cu comanda DOS FDISK, dar cu mult mai multe capabilități în comparație cu versiunea DOS.



Referință încrucișată

În acest capitol, vom explica modul în care puteți utiliza Disk Druid pentru a pregăti și utiliza partițiile discului pentru instalarea sistemului Red Hat Linux. Despre **fdisk** veți putea învăța în capitolul 11.

Înainte de a lansa Disk Druid pentru a partiționa discul, trebuie să știți modul în care puteți referi unitățile și partițiile de disc în Linux. De asemenea, trebuie să înțelegeți noțiunile de *punct de montare* și *partiție de swap*. Următoarele trei secțiuni descriu mai întâi acești termeni, după care se trece la prezentarea utilitarului Disk Druid.

Numele discurilor în Linux

Primul pas este de a reține modul în care sistemul Linux se referă la diferitele discuri. Sistemul Linux tratează toate dispozitivele fizice ca fișiere, dispunând de fapt de câte un fișier pentru a reprezenta fiecare dispozitiv în parte. Aceste *fișiere de dispozitiv* sunt localizate în directorul `/dev`. Dacă nu sunteți familiarizat cu UNIX, încă nu cunoașteți convenția de numire a fișierelor în UNIX, pe care o veți învăța însă pe măsură ce utilizați sistemul Linux. Dacă știți modul de alcătuire pentru numele de fișiere în MS-DOS, veți realiza că numele de fișiere din Linux sunt similare, exceptând faptul că nu utilizează litere (cum ar fi A și C) pentru denumirea unităților, iar separatorul între numele directoarelor este caracterul slash (/), față de MS-DOS, unde acesta este backslash (\).

Datorită faptului că sistemul Linux tratează toate dispozitivele ca fișiere din directorul `/dev`, numele unităților de disc încep cu `/dev`. Tabelul 1-3 prezintă o listă a numelor unităților de hard-disc, respectiv a unităților de dischetă de care puteți avea nevoie.

Tabelul 1-3 Numele unităților de hard-disc și de dischetă

Numele	Descrierea
<code>/dev/hda</code>	Prima unitate IDE (în DOS, aceasta este de regulă unitatea C)
<code>/dev/hdb</code>	A doua unitate IDE
<code>/dev/sda</code>	Prima unitate SCSI
<code>/dev/sdb</code>	A doua unitate SCSI
<code>/dev/fd0</code>	Prima unitate de dischetă (unitatea A din DOS)
<code>/dev/fd1</code>	A doua unitate de dischetă (unitatea B din DOS)

Atunci când folosiți utilitarul Disk Druid din Red Hat sau programul **fdisk** din Linux pentru a pregăti partițiile Linux, trebuie să identificați unitatea de disc printr-un nume, cum este `/dev/hda`.



Sugestie

Atunci când Disk Druid sau **fdisk** afișează lista de partiții, numele acestora sunt de forma `/dev/hda1`, `/dev/hda2` etc. Sistemul Linux construiește numele fiecărei partiții adăugând numărul partiției (cuprins între 1 și 4, pentru a desemna cele patru partiții primare de pe o unitate de disc) la numele discului. Așadar, dacă singura unitate de hard-disc IDE are două partiții, veți observa că în programul de instalare acestea sunt numite `/dev/hda1`, respectiv `/dev/hda2`. În continuare, iată mai multe exemple de nume pentru partițiile hard-discului în Linux:

Numele	Partiția
<code>/dev/hda1</code>	Prima partiție primară a primei unități IDE
<code>/dev/hda2</code>	A doua partiție primară a primei unități IDE
<code>/dev/hda3</code>	A treia partiție primară a primei unități IDE
<code>/dev/hda4</code>	A patra partiție primară a primei unități IDE
<code>/dev/hda5</code>	Prima partiție logică a primei unități IDE
<code>/dev/hda6</code>	A doua partiție logică a primei unități IDE
<code>/dev/hdb1</code>	Prima partiție primară a celei de-a doua unități IDE

(continuare)

Numele	Partiția
/dev/sda1	Prima partiție primară a primei unități SCSI
/dev/sda2	A doua partiție primară a primei unități SCSI
/dev/sdb1	Prima partiție primară a celei de-a doua unități SCSI
/dev/sdc1	Prima partiție primară a celei de-a treia unități SCSI

Punct de montare

În sistemul Linux, partițiile hard-diskului sunt utilizate prin asocierea cu o anumită parte a sistemului de fișiere, care este o organizare ierarhică de directoare – un arbore de directoare. Dacă aveți mai multe partiții de disc (este posibil ca partițiile Linux să se găsească și pe un al doilea disc), acestea pot fi toate utilizate în Linux. Este suficient să stabiliți ce parte a arborelui de directoare din Linux va fi localizată pe fiecare dintre partiții – operație care în Linux este cunoscută sub denumirea de *montare a unui sistem de fișiere într-un dispozitiv* (dispozitivul este în acest caz partiția discului).



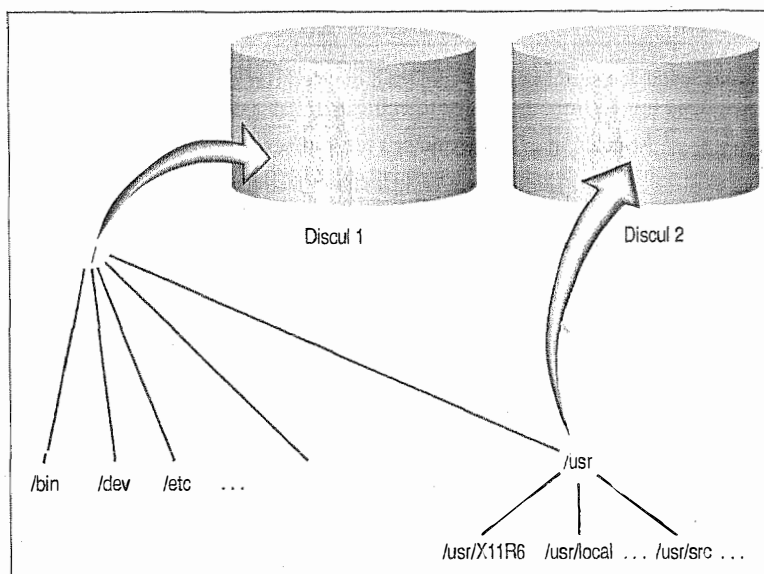
Observație

Termenul de punct de montare se referă la directorul care este asociat cu o partiție de disc sau cu un alt dispozitiv fizic.

Să presupunem că pe PC-ul dumneavoastră dispuneți de două discuri și ați creat partiții Linux pe ambele. Figura 1-14 prezintă modul în care puteți monta diferite părți ale arborelui de directoare din Linux (adică ale sistemului de fișiere) în aceste două partiții.

Figura 1-14

Un exemplu de montare a sistemului de fișiere Linux pe două partiții ale discului.



Partiția de swap

Majoritatea sistemelor de operare avansate implementează conceptul de memorie virtuală, care presupune utilizarea unui spațiu de pe discul sistemului ca extensie a memoriei fizice (RAM).

Atunci când sistemul de operare rămâne fără memorie disponibilă, poate deplasa conținutul unor zone inutil ocupate ale memoriei RAM, creând spațiul necesar pentru un program care are nevoie de mai multă memorie. De îndată ce este necesar accesul la informația deplasată pe disc, sistemul de operare caută alte informații pe care le deplasează pe disc pentru a aduce datele necesare înapoi în memorie. Acest proces de deplasare a informației din memoria RAM pe disc și invers este cunoscut sub denumirea de *paginare*.

Din cauza faptului că discul este mult mai lent decât memoria RAM, performanțele scad simțitor atunci când sistemul de operare efectuează multe operații de paginare; în schimb, memoria virtuală permite rularea unor programe pentru care altfel resursele sistemului ar fi fost insuficiente.

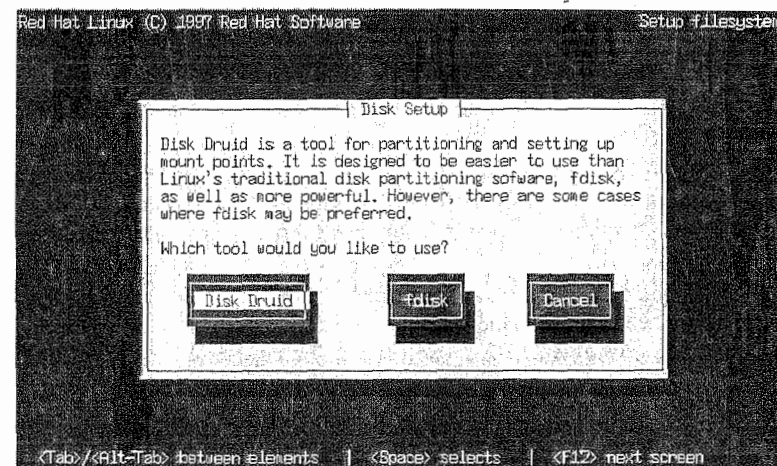
Sistemul Linux oferă suport pentru mecanismul de memorie virtuală și poate de asemenea utiliza o partiție de swap. Atunci când creați partițiile Linux, este necesar să creați și o partiție de swap. Cu utilitarul Disk Druid, prezentat în secțiunea următoare, crearea unei partiții de swap este o operație cât se poate de simplă. Marcați pur și simplu tipul unei partiții ca fiind dispozitiv de swap și programul Disk Druid va executa toate operațiile necesare.

Pregătirea discului

Atunci când sunteți gata să pregătiți discul pentru instalarea sistemului Linux, va apărea caseta de dialog Disk Setup, cu butonul Disk Druid deja selectat, după cum se poate observa în figura 1-15. Pentru a utiliza Disk Druid, apăsați tasta Enter.

Figura 1-15

Caseta de dialog Disk Setup din programul de instalare al sistemului Red Hat Linux



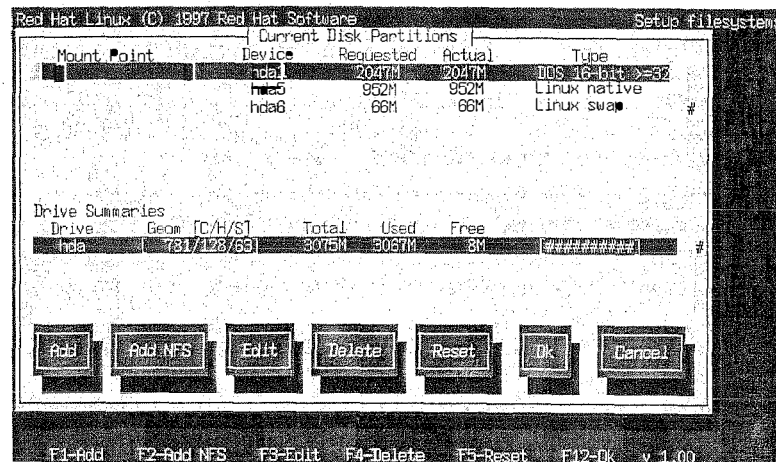
Programul Disk Druid colectează informația despre unitățile de hard-disk din sistem și afișează o casetă de dialog cu lista acestor unități și informația curentă de partiționare pentru una dintre aceste unități, după cum se poate observa în figura 1-16.

În programul Disk Druid, veți efectua operații specifice de configurare a discului, utilizând cele șapte butoane situate la baza casetei de dialog. În mod particular, butoanele permit efectuarea următoarelor acțiuni:

■ **Add** vă permite să creați o nouă partiție, presupunând că există suficient spațiu disponibil pe disc pentru a crea partiția. Atunci când apăsați acest buton, va apărea o altă casetă de dialog în care veți completa informația necesară pentru a crea o partiție.

Figura 1-16

Caseta principală de dialog din programul Disk Druid



- **Add NFS** este utilizat pentru asocierea unui director NFS – un director localizat pe un alt sistem din rețea – cu un punct de montare din sistemul dumneavoastră. Directorul NFS este montat numai cu drept de citire.
- **Edit** vă permite să modificați atributele unei partiții deja existente. Modificările atributelor curente le veți efectua prin intermediul unei alte casete de dialog, care apare la apăsarea butonului Edit.
- **Delete** se utilizează pentru ștergerea partiției curent marcate din lista Current Disk Partitions (vezi figura 1-16).
- **Reset** reface tabela de partiții așa cum era înainte de a lansa Disk Druid. Acest lucru este posibil datorită faptului că Disk Druid nu modifică tabela de partiții (care este păstrată pe hard-disc) decât atunci când părăsiți programul prin apăsarea butonului OK.
- **OK** determină afișarea unui mesaj prin care Disk Druid vă întreabă dacă doriți într-adevăr să actualizați tabelele de partiții de pe toate unitățile de hard-disc din sistem. Dacă răspundeți cu Yes, Disk Druid modifică tabelele de partiții și apoi se încheie.
- **Cancel** se utilizează pentru a părăsi programul Disk Druid fără a salva nici una dintre modificările efectuate.

Acțiunile pe care le veți executa efectiv utilizând Disk Druid depind în mare parte de unitățile de disc de care dispuneți în configurație și de partițiile deja configurate. În cazul de față, vom presupune că ați creat deja spațiul pe disc necesar pentru Linux prin una dintre următoarele metode:

- Ați pornit de la o singură unitate cu o singură partiție (numai unitatea C în Windows). Ați utilizat apoi FIPS pentru a despărți acea partiție în două (vezi secțiunea „Repartiționarea utilizând FIPS“). După repartiționare, vor fi două partiții DOS.
- PC-ul dumneavoastră avea o singură unitate de disc, suficient de mare (cel puțin 2 GB) cu două partiții (unitățile C și D din Windows). În acest caz, cea de-a doua partiție este o partiție extinsă care conține unitatea logică D. Sistemul Linux se va instala pe partiția extinsă, care constituia unitatea D în Windows.

Ambele situații presupun efectuarea aceleiași secvențe de pași:

1. Ștergerea partiției DOS – cea care dispune de spațiu suficient pentru instalarea sistemului Linux, nu cea ocupată de Windows.

2. Crearea în spațiul disponibil pe disc a două partiții Linux – una ca spațiu de swap, iar cealaltă pentru sistemul de fișiere Linux.

Următoarea secțiune vă prezintă în detaliu pașii pe care trebuie să-i efectuați utilizând programul Disk Druid pentru a configura numărul minim de partiții de care sistemul Linux are nevoie. Puteți pregăti și alte partiții de disc (dacă dispuneți de a doua unitate de disc, spre exemplu) urmând pașii pe care-i vom evidenția în secțiunea următoare.



Referință încrucișată

După ce configurați partițiile, trebuie să inițializați partiția de swap și să formatați partiția destinată pentru sistemul de fișiere Linux. Acești pași sunt prezentați în secțiunile „Inițializarea spațiului de swap“, respectiv „Formatarea partițiilor“.

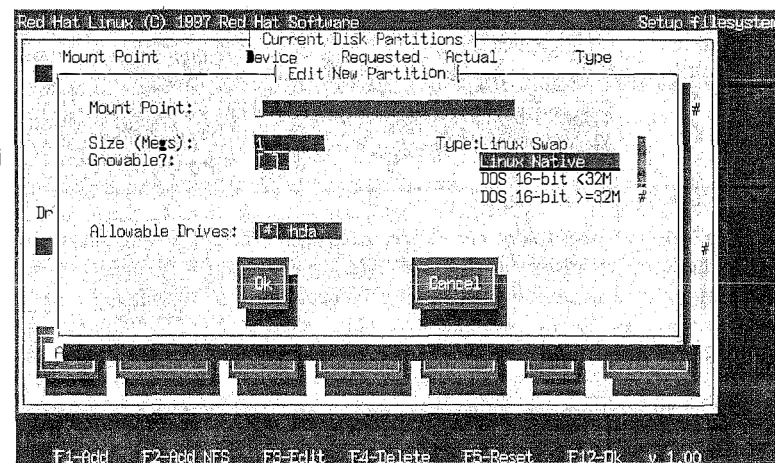
Configurarea partițiilor

Pentru pregătirea unei partiții DOS existente pentru Linux, trebuie să executați următorii pași în programul Disk Druid:

1. Ștergeți partiția DOS pe care doriți să o utilizați pentru Linux. Pentru aceasta, selectați partiția (aceasta va avea în majoritatea cazurilor numele de dispozitiv hda2) din lista Current Disk Partitions (vezi figura 1-16) și apoi apăsați butonul Delete. Același rezultat îl puteți obține mai rapid selectând partiția și apăsând tasta **F4**. Înainte de a șterge o partiție, amintiți-vă să notați pe hârtie dimensiunea partiției, afișată în lista Current Disk Partitions.
2. Creați o nouă partiție pentru sistemul de fișiere Linux. Pentru aceasta, apăsați butonul Add (sau tasta **F1**). Va apărea caseta de dialog Edit New Partition (vezi figura 1-17), unde trebuie să introduceți / ca punct de montare și dimensiunea partiției în megabaiți. Pentru a calcula dimensiunea, este suficient să scădeți dimensiunea spațiului de swap (32 MB sau cantitatea de memorie RAM instalată pe PC-ul dumneavoastră, dacă aceasta depășește 32 MB) din dimensiunea inițială a partiției. Selectați butonul OK și apoi apăsați tasta **Enter** pentru a termina acest pas și a vă întoarce la caseta de dialog principală a programului Disk Druid.

Figura 1-17

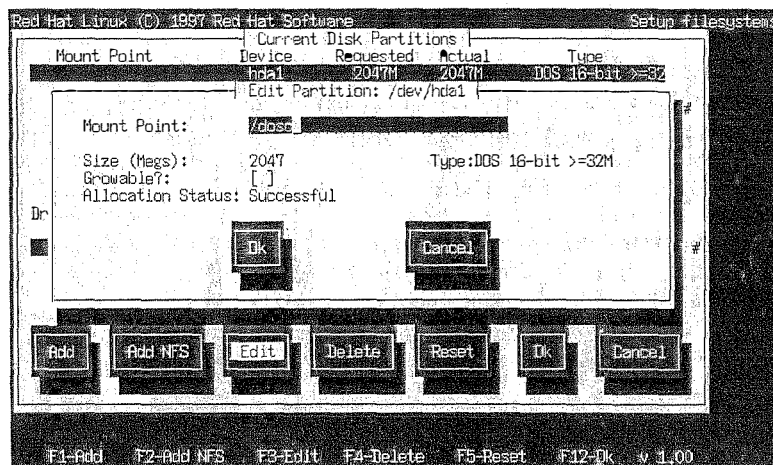
Caseta de dialog Edit New Partition, în care introduceți atributele unei noi partiții



3. Creați o altă nouă partiție și configurați-o ca spațiu de swap pentru Linux. Pentru aceasta, apăsați tasta F1 din caseta de dialog principală a programului Disk Druid (vezi figura 1-16). Apoi, în caseta de dialog Edit New Partition (vezi figura 1-17), introduceți dimensiunea partiției. Apăsați tasta Tab pentru a vă deplasa până la lista de tipuri de partiții și selectați cu ajutorul tastelor direcționale tipul Linux Swap – după aceasta, în câmpul Mount Point va apărea textul „Swap Partition“. Apăsați apoi tasta Tab pentru a selecta butonul OK și apăsați Enter pentru a defini noua partiție și a vă întoarce în caseta principală de dialog din programul Disk Druid.
4. Dacă doriți să aveți acces la partiția DOS din Linux, asigurați-vă să stabiliți un punct de montare pentru această partiție. Aceasta înseamnă să stabiliți numele directorului din Linux în care va apărea partiția DOS. Spre exemplu, puteți utiliza /dosc ca punct de montare pentru partiția DOS; fișierele DOS vor fi ulterior accesibile în directorul /dosc din Linux. Aceasta este o mnemonică bună, deoarece numele dosc vă va aminti că aceasta este chiar unitatea C din DOS. Pentru atribuirea unui punct de montare, selectați partiția DOS (de regulă, aceasta apare cu numele de dispozitiv hda1 în lista Current Disk Partitions) și apăsați F3 pentru a modifica atributele. Va apărea caseta de dialog Edit Partition (vezi figura 1-18). Introduceți punctul de montare (spre exemplu, /dosc) și apăsați Enter pentru a vă întoarce în caseta principală de dialog din programul Disk Druid.
5. Confirmați modificările efectuate apăsând butonul OK din caseta principală de dialog a programului Disk Druid (figura 1-16).

Figura 1-18

Caseta de dialog Edit Partition vă permite să modificați atributele unei partiții existente



Inițializarea spațiului de swap

După ce terminați operația de specificare a partițiilor în Disk Druid, programul de instalare a sistemului Red Hat afișează o casetă de dialog (vezi figura 1-19) care cuprinde o listă a tuturor partițiilor cu tipul Linux Swap și vă întreabă dacă doriți să inițializați spațiile de swap. Dacă ați creat o singură partiție de swap, lista va conține doar acea partiție. Continuați și inițializați partiția de swap apăsând tasta Tab până când selectați butonul OK, după care apăsați Enter.

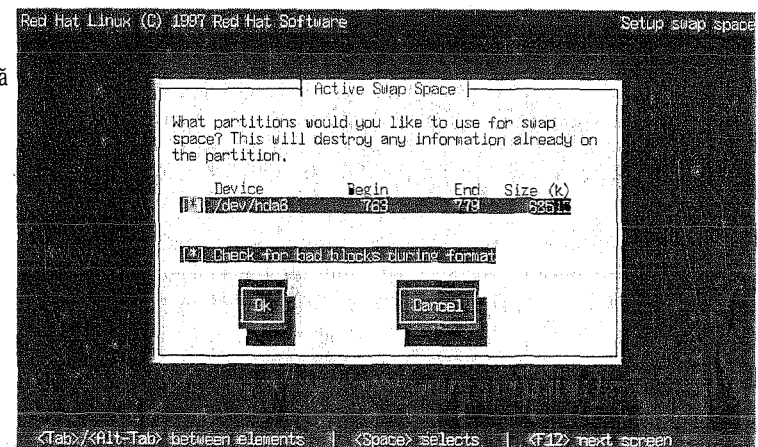


Sugestie

Dacă programul de instalare nu găsește nici o partiție de swap, este posibil să fi uitat să stabiliți tipul uneia dintre partiții ca fiind Linux Swap.

Figura 1-19

Caseta de dialog care vă permite să inițializați spațiul de swap

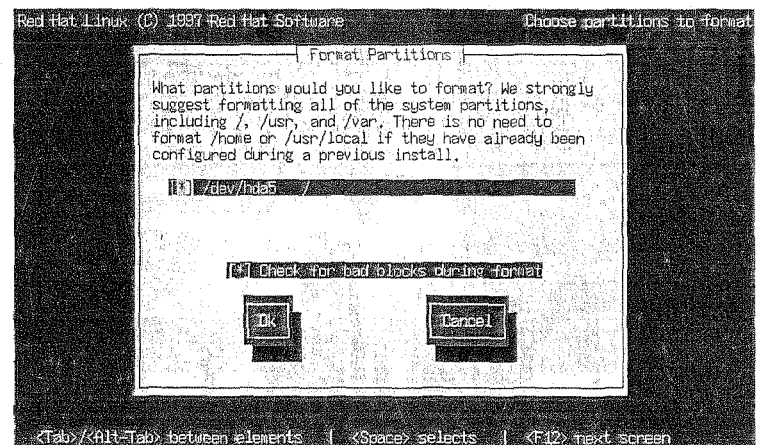


Formatarea partițiilor

După ce programul de instalare a sistemului Red Hat inițializează spațiul de swap, afișează o casetă de dialog care cuprinde o listă a partițiilor care trebuie formate pentru a putea fi utilizate în sistemul Linux. Dacă ați creat o singură partiție de disc pentru Linux, lista va cuprinde doar acea partiție (vezi figura 1-20).

Figura 1-20

Caseta de dialog Format Partitions vă permite să formatați partițiile Linux



Pentru a formata partiția, apăsați tasta Tab pentru a deplasa cursorul la numele partiției, după care apăsați tasta Spacebar pentru a selecta partiția. Trebuie de asemenea să selectați caseta marcată cu Check for bad blocks during format, astfel încât operația de formatare să marcheze toate zonele de pe disc care prezintă defecte fizice.



Observație

Dacă aveți mai multe partiții de disc montate în directoare diferite ale sistemului de fișiere Linux și efectuați actualizarea unei instalări mai vechi, nu mai este necesar să formatați partițiile în care doriți să păstrați informațiile existente. De exemplu, dacă toate directoarele utilizatorilor sistemului se găsesc pe o partiție separată montată în directorul /home, nu trebuie să formatați și acea partiție.

Selecția componentelor pentru instalare

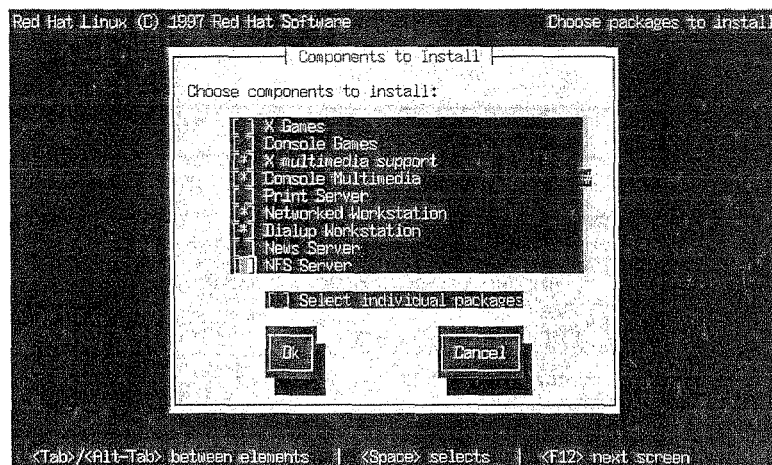
După ce ați selectat partițiile care vor fi formate, programul de instalare nu trece imediat la formatarea acestora. Înainte de aceasta, programul vă cere să specificați componentele sistemului Red Hat Linux pe care doriți să le instalați. În acest mod, după ce ați selectat componentele vă puteți bucura de o ceașcă de cafea în timp ce programul de instalare a sistemului Red Hat va formata partițiile selectate și va copia toate fișierele necesare pe acele partiții.

Sistemul Red Hat utilizează fișiere speciale, numite *pachete* (package), pentru a împacheta un număr de fișiere care intră în alcătuirea anumitor programe. Spre exemplu, toate fișierele de configurare, documentația și fișierele binare pentru limbajul de programare Perl vin „ambalate” într-un pachet Red Hat. Pentru a instala, dezinstala sau obține informații despre pachete se utilizează un program numit Red Hat Package Manager (RPM). Utilizarea programului RPM este prezentată în capitolul 2. Deocamdată, este suficient să rețineți că o componentă a sistemului de operare este alcătuită de regulă din mai multe pachete Red Hat.

Figura 1-21 prezintă caseta de dialog cu lista componentelor pe care le puteți alege pentru instalare. Fiecare componentă este identificată printr-o etichetă descriptivă, prefixată de o pereche de paranteze pătrate ([]).

Figura 1-21

Caseta de dialog din care puteți selecta componentele pe care le veți instala



Prezența unui asterisc (*) în interiorul parantezelor pătrate indică faptul că respectiva componentă este selectată. Atunci când caseta de dialog apare pentru prima dată, multe dintre componente sunt deja selectate, acestea fiind indicate prin asteriscuri. Puteți gândi setul componentelor marcate ca fiind setul minimal de componente recomandat pentru instalarea sistemului Red Hat. Puteți însă opta pentru instalarea oricăror altor componente sau chiar a tuturor componentelor. Utilizați tastele direcționale pentru a vă deplasa în lista acestora (figura 1-21) și apăsați tasta Spacebar pentru a selecta sau deselecta o componentă.

Tabelul 1-4 prezintă componentele din această listă. În tabel, prezența unui asterisc indică o componentă selectată în mod prestabilit. Pe lângă aceste componente care pot fi selectate de către utilizatori, programul de instalare a sistemului Red Hat va instala automat un număr mare de pachete necesare pentru rularea sistemului Linux.



Sugestie

Fiecare componentă necesită anumite pachete pentru a putea rula. Programul de instalare a sistemului Red Hat verifică dependențele între diferitele pachete și vă prezintă o listă a pachetelor necesare, dar pe care nu le-ați selectat. În această situație, se recomandă să instalați și aceste pachete necesare.

Cum fiecare componentă reprezintă o colecție de pachete distincte, programul de instalare vă oferă de asemenea posibilitatea de a selecta în mod individual diferitele pachete. Dacă marcați caseta numită „Select individual packages” care apare la sfârșitul listei din figura 1-21 și apoi apăsați pe butonul OK, programul de instalare vă prezintă în continuare alte casete de dialog din care puteți selecta diferitele pachete în mod individual. Dacă însă sunteți la prima instalare a sistemului Linux, nu este necesar să intrați în detalii până la nivelul de instalare a pachetelor în mod individual. Este suficient să alegeți din tabelul 1-4 componentele de care considerați că aveți nevoie. Este posibil să instalați ulterior pachetele de care aveți nevoie cu ajutorul utilitarului RPM.

Tabelul 1-4 Lista componentelor ce pot fi instalate în sistemul Red Hat Linux

Numele componentei	Descriere
Printer Support	Pachete necesare pentru a tipări la imprimantă. Include utilitare cum ar fi lpr sau Ghostscript.
*X Window System	Pachete care alcătuiesc XFree86 3.3.1, sistemul X Window pentru calculatoare bazate pe procesoare Intel x86. Include toate fonturile, programul de gestiune a ferestrelor <i>fvwm2</i> , Ghostscript, Tcl/Tk, Perl și multe programe utilitare. Pachetul care alcătuiește efectiv serverul X este selectat cu ajutorul unui program de configurare care este rulat într-o etapă ulterioară a instalării.
*Mail/WWW/News Tools	Pachete cum sunt <i>elm</i> , <i>pine</i> și <i>trn</i> pentru citirea poștei electronice și a grupurilor de discuții. Componenta include multe alte pachete pentru sistemul X Window, Perl, Tcl/Tk sau care asigură suportul necesar imprimării.
DOS/Windows Connectivity	Pachete cum sunt <i>dosemu</i> , <i>mttools</i> , <i>lha</i> , <i>zip</i> și <i>unzip</i> , utile pentru accesul la fișiere DOS sau Windows din sistemul Linux. Tot aici se găsesc alte multe pachete legate de sistemul X Window.
*File Managers	Programe de gestiune a fișierelor, cum sunt <i>mc</i> (Midnight Commander) și <i>xfm</i> , care necesită limbajele Perl, Tcl/Tk și sistemul X Window.
Graphics Manipulation	Pachete care permit manipularea imaginilor, cum sunt ImageMagick, <i>gimp</i> și Ghostscript, care rulează sub sistemul X Window.
X Games	Întreg sistemul X Window, împreună cu pachetele <i>xbill</i> , <i>xboard</i> , <i>xboing</i> , <i>xchomp</i> , <i>xdemineur</i> , <i>xevil</i> , <i>xfishtank</i> , <i>xgalaga</i> , <i>xgammon</i> , <i>xgopher</i> , <i>xjewel</i> , <i>xlander</i> , <i>xpat2</i> , <i>xpilot</i> , <i>xpuzzles</i> , <i>xsnow</i> și <i>xtrojka</i> .
Console Games	Jocuri UNIX vechi, ca de exemplu <i>hangman</i> sau <i>fish</i> , precum și unele mai noi, cum ar fi <i>yahtzee</i> , <i>Doom</i> , <i>gnuchess</i> , <i>trojka</i> , precum și câteva jocuri de cărți.
*X Multimedia Support	Pachete pentru utilizarea sunetelor și animației în sistemul X. Include de asemenea toate pachetele sistemului X Window.

(continuare)

Tabelul 1-4 Continuare

Numele componentei	Descriere
*Console Multimedia	Pachete care permit lucrul cu sunete și animație cum ar fi aumix, sndconfig, rlsound, cdp, maplay și playmidi.
Print Server	Pachete necesare pentru a transforma sistemul Linux într-un server de imprimare.
*Networked Workstation	Pachete care oferă serviciile rețelelor TCP/IP, cum sunt FTP (File Transfer Protocol), Telnet și NFS (Network File System).
*Dialup Workstation	Pachete care oferă servicii TCP/IP, precum și pachete cum sunt ppp și dip, necesare pentru stabilirea unei conexiuni prin liniile telefonice. Tot aici se găsesc și pachete de comunicație, cum sunt minicom și lrzsz (utilitar pentru trimiterea și recepționarea fișierelor).
News Server	Pachetele Perl și INN (Internet News server), necesare pentru configurarea sistemului Linux ca server pentru grupurile de discuții.
NFS Server	Pachete cum sunt portmap și nfs-server, care permit configurarea sistemului Linux ca server NFS.
SMB (Samba) Connectivity	Pachete cum sunt smb și smbfs, necesare pentru utilizarea PC-ului cu sistemul Linux ca server LAN Manager. Tot aici sunt incluse multe alte pachete, printre care cele pentru rețele TCP/IP și cele necesare lucrului cu imprimanta.
IPX/Netware (tm) Connectivity	Pachete necesare pentru utilizarea sistemului Linux ca server Novell Netware. Componenta include de asemenea multe dintre pachetele pentru rețelele TCP/IP.
Anonymous FTP/Gopher Server	Pachetele wu-ftp și anonftp, împreună cu toate pachetele pentru rețele TCP/IP, necesare pentru a oferi serviciul de FTP Anonymous în Linux.
Web Server	Severul de Web Apache împreună cu alte pachete pentru rețele, necesare pentru utilizarea sistemului Linux ca server Web.
DNS Name Server	Pachetele bind, bind-utils și caching-nameserver, împreună cu altele necesare pentru a oferi serviciul Domain Name Service (DNS) pe sistemul Linux.
Postgres (SQL) Server	Pachetele postgresql, postgresql-data și postgresql-devel, împreună cu multe altele necesare pentru implementarea serverului Postgres SQL (pentru baze de date) pe sistemul Linux.
Network Management Workstation	Pachete care oferă facilitățile de gestiune a rețelei prin SNMP (Simple Network Management Protocol) și utilitarul de administrare a firewall-urilor IP. Componenta include multe alte pachete necesare instrumentelor de gestiune a rețelei.
TeX Document Formatting	Pachete necesare pentru TeX (se pronunță tek), sistem de formatare a documentelor.
Emacs	Versiune a editorului de texte Emacs care nu necesită sistemul X.
Emacs with X Windows	GNU Emacs 20.2, care rulează sub X. Componenta include de asemenea sistemul X Window, ca și versiunea de Emacs care nu necesită X.

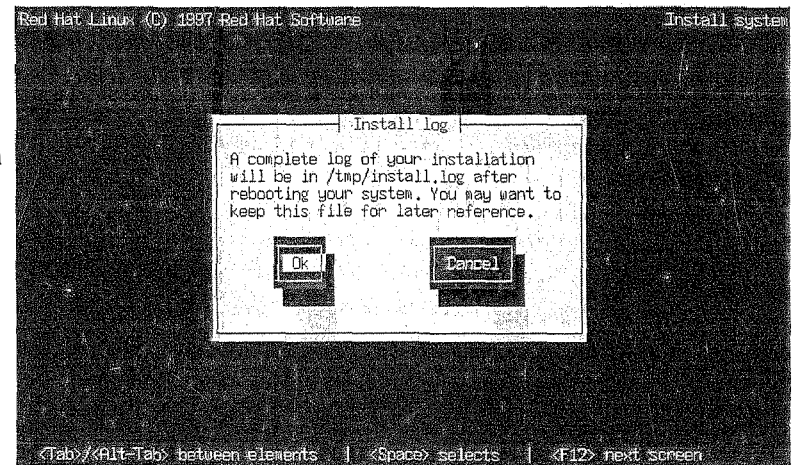
Numele componentei	Descriere
C Development	Compilatorul de C gcc, depanatorul gdb, utilitarul Revision Control System (RCS) și alte utilitare necesare pentru programarea în C.
Development Libraries	Diferite biblioteci necesare pentru dezvoltarea de programe în Linux.
C++ Development	Compilatorul GNU pentru limbajul C++, împreună cu alte fișiere necesare pentru programarea în C++.
X Development	Fișierele antet și bibliotecile de funcții X, precum și sistemul X Window, necesare pentru dezvoltarea aplicațiilor care utilizează X.
Extra Documentation	Diferite manuale online (denumite pagini man în UNIX), precum și diferite fișiere HOWTO pentru Linux și fișiere cu întrebările cele mai frecvente (Frequently Asked Questions – FAQ) pentru multe dintre programele din Linux. Este recomandat să instalați această componentă.
Everything	Selectați acest element pentru a instala toate componentele.

Componentele marcate printr-un asterisc () sunt selectate în mod prestabilit.

După ce ați selectat componentele pe care doriți să le instalați, apăsați tasta Tab pentru a vă deplasa pe butonul OK din figura 1-21 și apăsați tasta Enter. Programul de instalare va afișa o casetă de dialog informativă, după cum se poate observa și în figura 1-22.

Figura 1-22

Mesaj informativ despre raportul asupra instalării, afișat înainte de instalarea efectivă a componentelor selectate



Casetă de dialog vă informează că în fișierul /tmp/install.log se va găsi un raport al procesului de instalare. În acest fișier se va găsi o listă a pachetelor Red Hat instalate în sistemul dumneavoastră. Iată primele câteva linii ale fișierului /tmp/install.log după o instalare normală a sistemului Red Hat Linux pe un PC:

```
Installing setup.
Installing filesystem.
Installing basesystem.
Installing anonftp.
Installing AnotherLevel.
Installing ldconfig.
Installing termcap.
Installing libtermcap.
```

```
Installing glibc.
Installing grep.
Installing fileutils.
Installing bash.
Installing aout-libs.
Installing slang..
Installing newt.
Installing chkconfig.
Installing apache.
```

Puteți vedea ulterior raportul asupra instalării, păstrând acest fișier pentru comparații viitoare.

Apăsați tasta Enter pentru a continua instalarea. Programul de instalare a sistemului Red Hat va formata partițiile discului și apoi va instala pachetele. Pe măsură ce pachetele sunt instalate, programul de instalare afișează pe un ecran starea curentă a procesului de instalare, oferind informații cum ar fi numărul total de pachete ce vor fi instalate, numărul instalat până în momentul curent, o estimare a spațiului necesar pe disc și a timpului rămas până la terminarea procesului.



Sugestie

Formatarea și instalarea sunt operații de durată – puteți lua o pauză, întorcându-vă să observați ce se întâmplă după aproximativ zece minute. Atunci când vă întoarceți, veți putea obține o estimare a timpului rămas până la finalul operației de pe ecranul de stare pe care programul de instalare îl actualizează continuu.

Configurarea sistemului Linux

După ce programul de instalare termină de încărcat toate pachetele selectate, urmează etapa de configurare. Presupunând că doriți să instalați sistemul X Window și componenta de rețea, pașii etapei de configurare sunt următorii:

- configurarea mouse-ului
- configurarea sistemului X
- configurarea rețelei
- stabilirea fusului orar
- configurarea serviciilor
- configurarea imprimantelor
- stabilirea parolei de root
- crearea unei dischete de pornire
- instalarea programului LILO

Secțiunile următoare prezintă fiecare dintre acești pași de configurare.

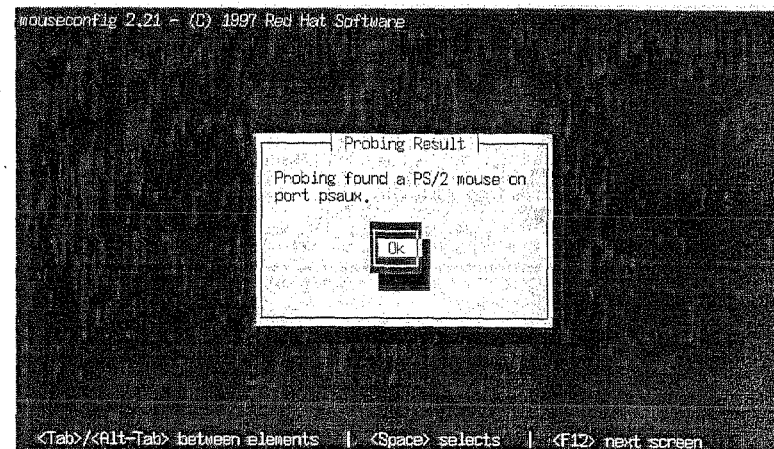
Configurarea mouse-ului

Dacă ați instalat componenta X Window System (operație descrisă în secțiunea „Alegerea componentelor pentru instalare”), programul de instalare a sistemului Red Hat rulează utilitarul `mouseconfig` pentru a detecta tipul mouse-ului din configurația dumneavoastră. Figura 1-23 prezintă un rezultat tipic al operației de detecție a mouse-ului.

În acest caz, programul `mouseconfig` a detectat un mouse care utilizează o interfață de tip PS/2. Acest mesaj are doar un scop informativ; apăsați deci Enter pentru a trece la pasul următor.

Figura 1-23

Casetă de dialog care vă precizează tipul de mouse detectat

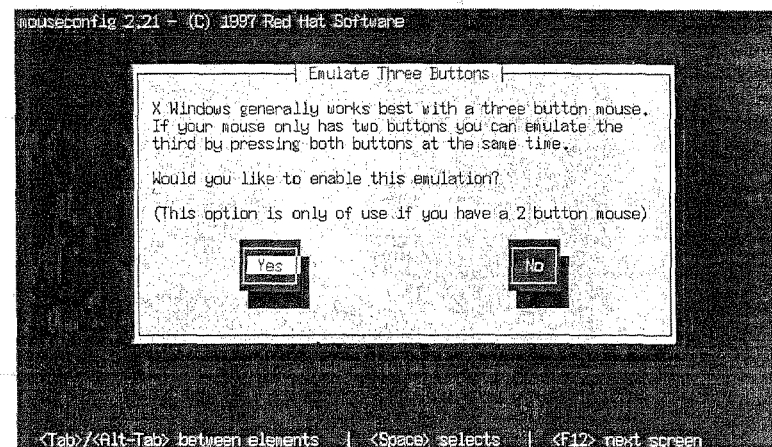


Programul de configurare a mouse-ului va afișa o altă casetă de dialog (vezi figura 1-24), care vă oferă posibilitatea de a emula un mouse cu 3 butoane folosind unul obișnuit, cu numai două butoane.

Dacă selectați Yes, puteți simula butonul din mijloc al mouse-ului prin apăsarea simultană a celor două butoane. Cum multe aplicații X utilizează un mouse cu 3 butoane, este recomandat să continuați selectând butonul Yes din caseta de dialog.

Figura 1-24

Casetă de dialog care vă cere să confirmați emularea unui mouse cu 3 butoane



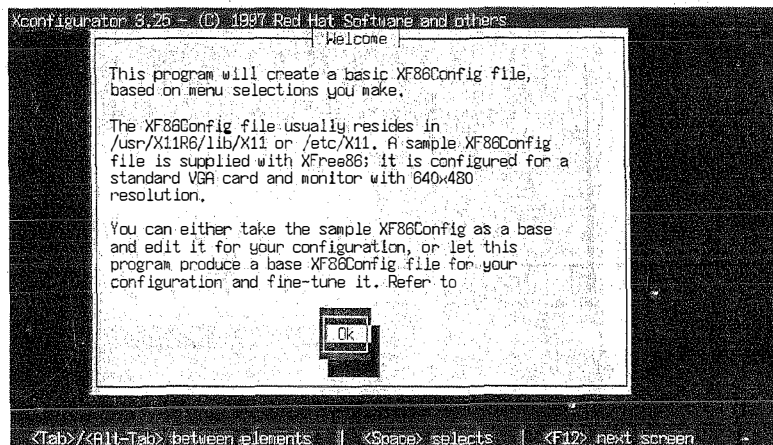
Configurarea sistemului X

Dacă ați selectat componenta X Window System (operație descrisă în secțiunea „Alegerea componentelor pentru instalare”), programul de instalare a sistemului Red Hat va rula acum utilitarul `xconfigurator`, pentru a genera un fișier de configurare necesar serverului X.

Programul `xconfigurator` începe prin afișarea unei casete de dialog introductive, prezentată în figura 1-25.

Figura 1-25

Casetă de dialog introductivă din programul Xconfigurator

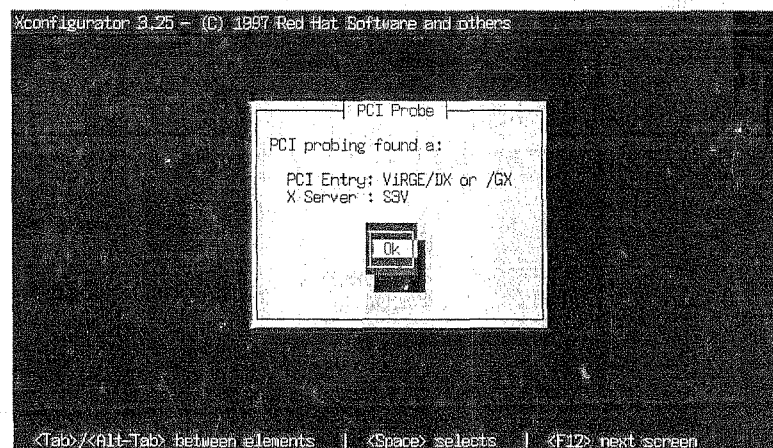


Pentru a citi și restul mesajului afișat în casetă, apăsați tasta cu săgeată în jos pentru a determina defilarea ecranului. După ce ați citit mesajul, apăsați tasta Enter pentru a continua operația de configurare.

Programul Xconfigurator detectează automat cipsetul utilizat de placa dumneavoastră video, afișând apoi un mesaj, după cum se observă în figura 1-26.

Figura 1-26

Rezultatul detectării plăcii video



În cazul de mai sus, programul Xconfigurator a detectat o placă video bazată pe cipsetul S3 ViRGE/DX (sau /GX), recomandând alegerea serverului X S3V. Mesajul afișat de Xconfigurator este oarecum dificil de înțeles. Serverul X S3V se referă de fapt la programul X Server cu numele complet XF86_S3V, care va fi instalat în directorul /usr/X11R6/bin. Cum caseta de dialog din figura 1-26 are doar un scop informativ, apăsați tasta Enter pentru a trece la restul configurării.

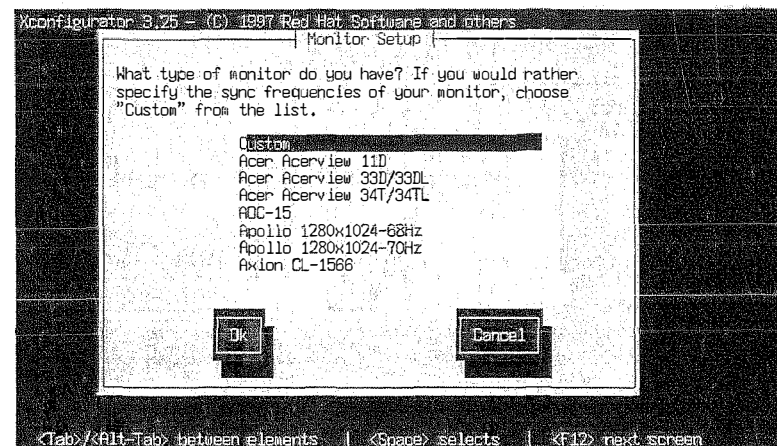
Următorul pas este selectarea monitorului dumneavoastră dintr-o listă, după cum se observă în figura 1-27.

Apăsați săgețile în sus și în jos pentru a vă deplasa prin listă. Dacă modelul monitorului dumneavoastră apare în listă, poziționați cursorul pe acel monitor și apoi apăsați tasta Enter. În caz contrar, alegeți opțiunea Custom și apăsați Enter.

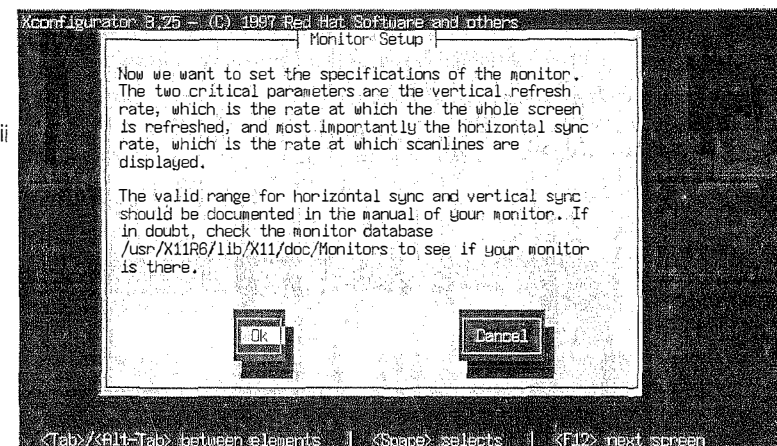
Dacă ați ales opțiunea Custom, programul Xconfigurator va afișa o altă casetă de dialog, după cum se observă în figura 1-28.

Figura 1-27

Selectarea tipului de monitor

**Figura 1-28**

Xconfigurator vă informează că are nevoie de anumite informații despre monitorul dumneavoastră



După cum precizează caseta de dialog (vezi figura 1-28), este nevoie să introduceți doi parametri esențiali ai monitorului dumneavoastră:

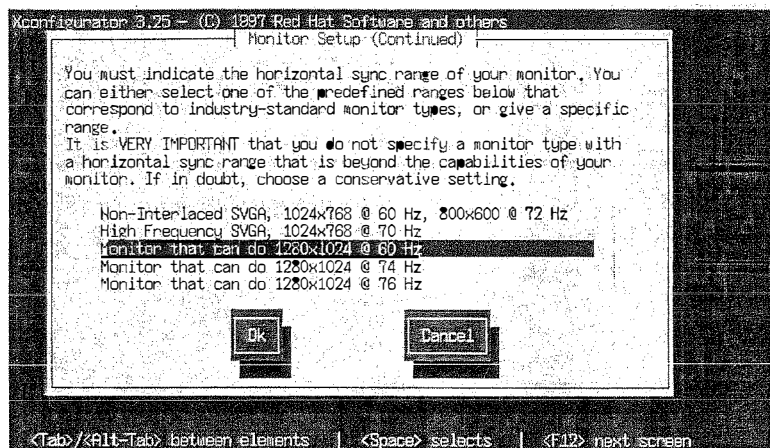
- frecvența de sincronizare pe orizontală – aceasta reprezintă numărul actualizărilor pe secundă pentru liniile de rastru orizontale și se exprimă în kilohertzi (kHz)
- rata de sincronizare pe verticală – aceasta exprimă de câte ori pe secundă monitorul poate actualiza întregul ecran

Aceste informații se pot găsi în manualul livrat împreună cu monitorul.

Apăsați tasta Enter ca răspuns la caseta de dialog din figura 1-28 pentru a trece mai departe. Programul Xconfigurator va afișa o listă cu caracteristici predefinite ale unor tipuri de monitoare, după cum se poate observa în figura 1-29.

Figura 1-29

Lista domeniilor predefinite pentru frecvențele de sincronizare pe orizontală afișate de către Xconfigurator



Atenție!

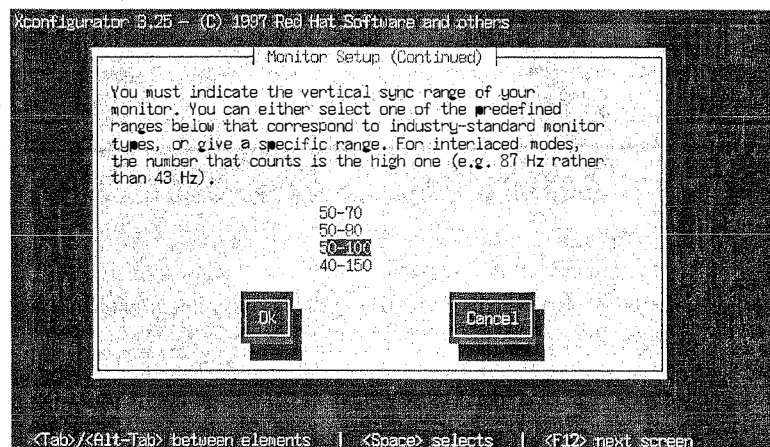
De regulă, unul din elementele din listă trebuie să corespundă cu parametrii monitorului dumneavoastră. Dacă sunteți în situația de a ghici parametrii monitorului, fiți precaut. Nu precizați un domeniu de sincronizare pe orizontală care să depășească posibilitățile monitorului. O valoare necorespunzătoare a frecvenței de sincronizare pe orizontală poate conduce la defectarea monitorului.

De exemplu, dacă în manualul monitorului scrie că acesta poate afișa o rezoluție de 1280 x 1024 la frecvența de 60 Hz, este necesar să alegeți opțiunea din listă care corespunde cu această specificație. După ce ați selectat domeniul de sincronizare pe orizontală, apăsați tasta Enter.

Programul Xconfigurator vă cere acum să introduceți rata de sincronizare pe verticală a monitorului, după cum se poate vedea și în figura 1-30.

Figura 1-30

Lista domeniilor de sincronizare pe verticală afișate de Xconfigurator

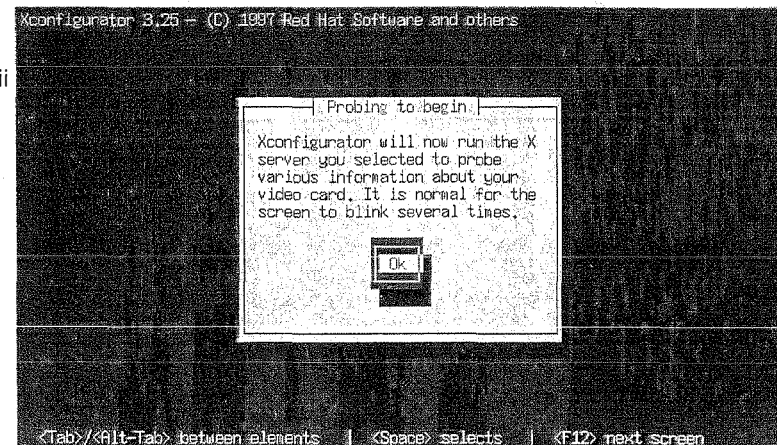


Alegeți domeniul de sincronizare pe verticală cel mai apropiat de specificațiile monitorului dumneavoastră și apăsați Enter.

Xconfigurator va afișa o casetă de dialog (vezi figura 1-31), informându-vă că va rula serverul X pentru a culege informații despre placa video.

Figura 1-31

Mesajul referitor la detectarea plăcii video de către Xconfigurator



Cum acest mesaj are un caracter pur informativ, citiți-l și apăsați apoi Enter pentru a continua.

Ecranul va clipi de câteva ori, după care Xconfigurator afișează un mod video prestabilit – rezoluția acestuia fiind 1024 x 768, iar culorile fiind reprezentate pe 8 biți. Pentru a accepta această valoare prestabilită, apăsați tasta Enter.

Programul Xconfigurator scrie apoi fișierul XF86Config în directorul /etc/X11 și afișează o casetă de dialog prin care vă informează că operația s-a terminat. Apăsați tasta Enter pentru a încheia etapa de configurare a serverului X.



Sugestie

După ce terminați instalarea și reporniți PC-ul pentru a reinițializa sistemul Linux, puteți încerca serverul X tastând startx la promptul Linux. Acum însă, continuați cu restul pașilor de configurare.

Configurarea rețelei

După ce ați terminat de configurat serverul X, programul de instalare a sistemului Red Hat va afișa o casetă de dialog (vezi figura 1-32) prin care vă întreabă dacă doriți să configurați parametrii rețelei locale (LAN) pentru sistemul dumneavoastră Linux.

Acest pas nu este destinat configurării unei conexiuni la rețea prin linie comutată (dial-up). Aveți nevoie să efectuați acest pas numai dacă sistemul Linux este conectat la o rețea TCP/IP printr-o placă Ethernet.

Dacă sistemul dumneavoastră este pe o rețea locală, chiar dacă acea rețea nu este conectată la Internet, este necesar să selectați butonul Yes. Programul de instalare a sistemului Red Hat va afișa o casetă de dialog în care trebuie să introduceți anumiți parametri pentru a configura protocolul TCP/IP (vezi figura 1-33).

Figura 1-32

Caseta de dialog care vă întreabă dacă doriți să configurați rețeaua locală

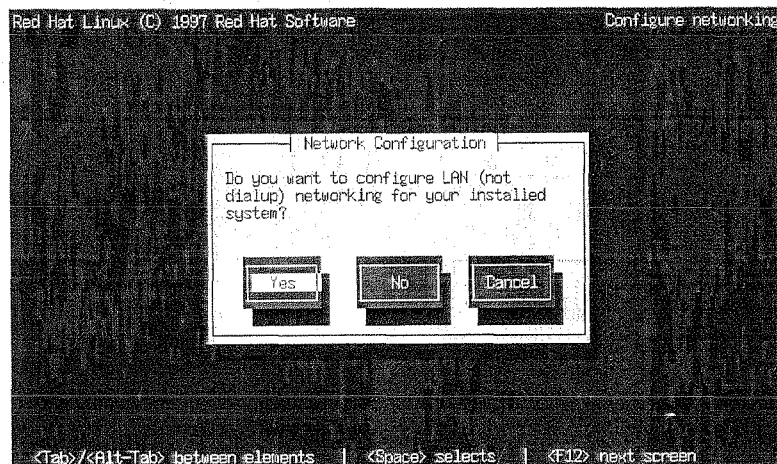
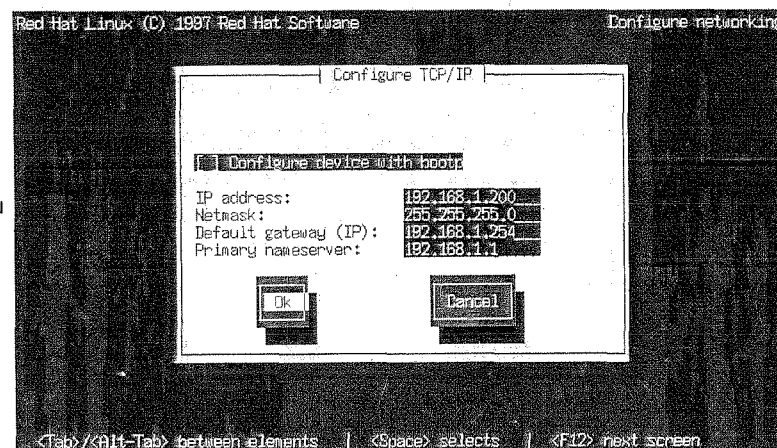


Figura 1-33

Caseta de dialog pentru configurarea protocolului TCP/IP, în care veți introduce adresele IP pentru rețeaua locală TCP/IP (valorile completate sunt ca model)



Caseta de dialog vă cere să introduceți patru parametri esențiali:

- adresa IP a sistemului Linux (cel pe care îl instalați în prezent)
- masca de rețea
- adresa IP a gateway-ului (sistemul prin intermediul căruia vă puteți conecta la orice rețea exterioară)
- adresa IP a serverului de nume.



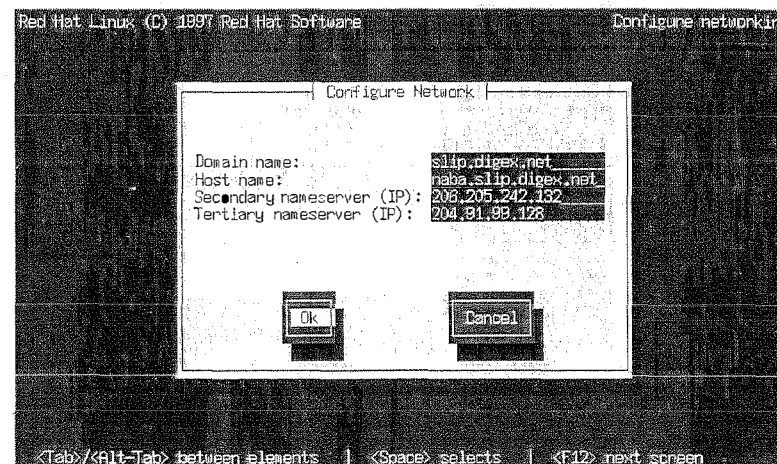
Referință încrucișată

Dacă rețeaua dumneavoastră este privată (adică nu este direct conectată la Internet), trebuie să utilizați o adresă IP dintr-un domeniu special rezervat pentru acest scop. Un domeniu de adrese des utilizat pentru rețelele private este între 192.168.1.1 și 192.168.1.254. Informații mai detaliate despre rețelele TCP/IP și adresele IP puteți găsi în capitolul 16.

După ce ați introdus parametrii ceruți, apăsați tasta Tab pentru a selecta OK, apoi apăsați Enter. Programul de instalare va afișa apoi o altă casetă de dialog (figura 1-34) prin care vă cere alte informații despre rețeaua locală TCP/IP.

Figura 1-34

Caseta de dialog pentru informații mai detaliate despre rețeaua locală (valorile completate au doar rolul de model)



Acum trebuie să introduceți următoarele informații:

- numele de domeniu al rețelei dumneavoastră
- numele sistemului dumneavoastră în cadrul rețelei (într-o rețea privată, puteți alege orice nume doriți)
- adresa IP a unui server secundar de nume
- adresa IP a unui server terțiar de nume

După ce ați completat informația cerută, selectați OK și apăsați tasta Enter.

Stabilirea fusului orar

După ce ați terminat cu configurarea rețelei, trebuie să alegeți fusul orar – adică diferența dintre ora locală și ora curentă la meridianul Greenwich, care este luat ca referință (această oră este cunoscută sub numele de Greenwich Mean Time sau GMT). Programul de instalare va afișa o listă de fusuri orare (vezi figura 1-35) corespunzătoare unor țări sau unor întregi regiuni geografice.

În listă apar fusurile orare pentru multe țări grupate pe continente, respectiv pe regiuni, pentru cazul Statelor Unite ale Americii. Utilizați săgețile în sus și în jos pentru a selecta poziția geografică în care vă aflați. Dacă de exemplu domiciliați în România, selectați Eastern Europe. Dacă țara în care locuiți nu apare în listă, selectați elementul care reprezintă diferența dintre ora locală și GMT. După ce ați selectat fusul orar, apăsați Enter.

Configurarea serviciilor

După selectarea fusului orar, programul de instalare Red Hat trece la stabilirea serviciilor care vor fi automat rulate la fiecare repornire a sistemului Linux. Serviciile dorite trebuie selectate din lista prezentată în figura 1-36.

Figura 1-35

Caseta de dialog din care puteți selecta fusul orar

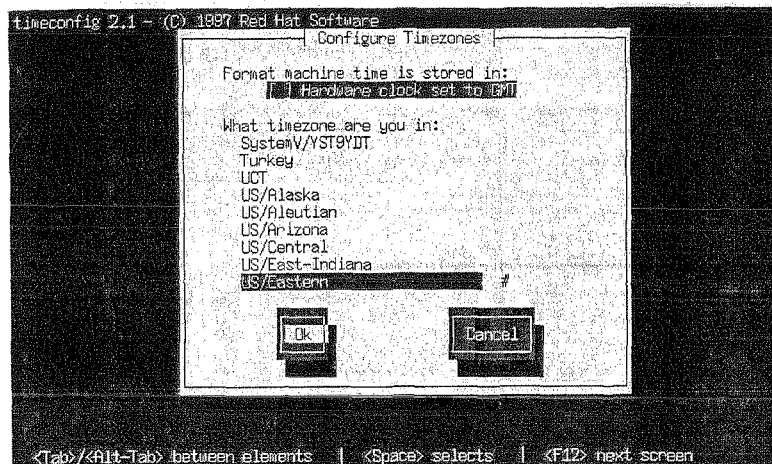
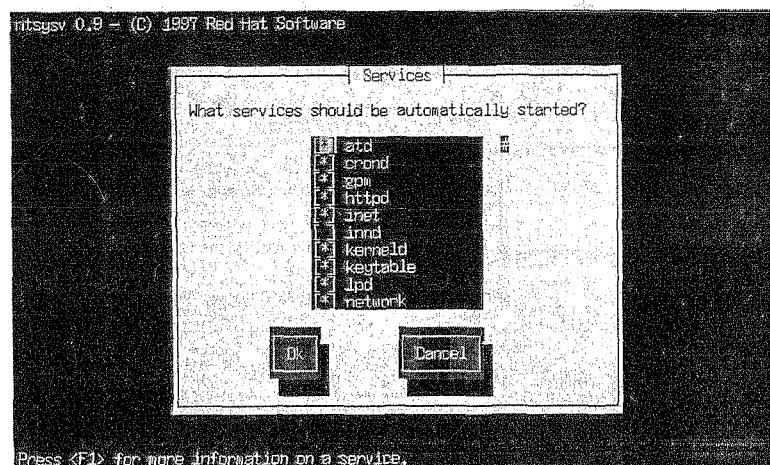


Figura 1-36

Caseta de dialog din care puteți selecta serviciile care vor fi rulate în mod automat



Fiecare linie din această listă reprezintă numele unui serviciu prefixat de o paranteză pătrată ([]). Prezența unui asterisc (*) în interiorul parantezei pătrate indică faptul că serviciul este selectat pentru a fi rulat în mod automat. Atunci când caseta de dialog apare pentru prima dată, multe dintre servicii sunt deja selectate. Puteți utiliza săgețile în sus și în jos pentru a vă deplasa în cadrul listei, apăsând tasta Spacebar pentru a selecta sau deselecta un anumit serviciu. După ce ați efectuat selecția, apăsați Tab pentru a selecta OK și apoi apăsați Enter.

Tabelul 1-5 prezintă o listă a serviciilor împreună cu o scurtă descriere a fiecăruia dintre acestea. Prima coloană conține numele serviciului, care coincide cu numele programului care trebuie rulat pentru a oferi acel serviciu. Numele care au un asterisc în coloana a doua sunt cele selectate în mod prestabilit de programul de instalare.

Tabelul 1-5 Servicii configurate pentru a porni automat în Linux

Numele serviciului	Selectat prestabilit	Descriere
atd	*	Rulează comenzile planificate de comenzile at și batch.
crond	*	Rulează programe precizate de utilizator, conform unui program periodic stabilit prin intermediul comenzii crontab.
gpm	*	Permite utilizarea mouse-ului pe terminalele alfanumerice.
httpd	*	Acesta este serverul de World Wide Web (WWW) Apache.
inet	*	Acesta este superserverul Internet, cunoscut și sub numele de inetd. Pornește celelalte servicii Internet, cum ar fi Telnet sau FTP, atunci când acestea sunt necesare.
innd		Serverul Internet News, util pentru a permite prezența unor grupuri locale de discuții pe sistemul dumneavoastră
kernel	*	Încarcă automat modulele nucleului sistemului Linux, pe măsură ce acestea sunt necesare. Acest serviciu trebuie întotdeauna rulat, deoarece sistemul Linux are nevoie de el.
keytable	*	Încarcă tipul selectat de tastatură specificat în fișierul /etc/sysconfig/keyboard. Acest serviciu trebuie rulat și în cadrul sistemului dumneavoastră
lpd	*	Server care gestionează coada de imprimare și trimite joburile către imprimantă. Aveți nevoie de acest server dacă doriți să utilizați orice tip de imprimantă din sistemul Linux.
mars-nwe		Acesta este un fișier MARS și un server de imprimare compatibil cu sistemul Novell Netware. Rulați acest serviciu numai dacă doriți să utilizați sistemul Linux ca server Netware.
named		Acesta este serverul de nume (Domain Name Sever, prescurtat DNS), care realizează conversia numelor calculatoarelor gazdă în adrese IP. Puteți rula o copie și pe sistemul dumneavoastră, dacă doriți.
network	*	Acest server vă permite să activați sau să dezactivați toate interfețele de rețea configurate pentru a fi utilizate în momentul pornirii sistemului.
nfs	*	Exportă sisteme de fișiere utilizând protocolul NFS (Network File System), astfel încât alte sisteme (care rulează și ele NFS) să poată partaja fișiere cu sistemul dumneavoastră
nfsfs	*	Server care permite montarea sau demontarea sistemelor de fișiere NFS.
portmap	*	Server utilizat de orice program bazat pe protocolul Remote Procedure Calls (RPC). Sistemul NFS necesită activarea serviciului portmap.
postgresql	*	Pornește și oprește serverul PostgreSQL, care gestionează cereri la o bază de date (PostgreSQL este un sistem gratuit de baze de date, livrat împreună cu sistemul Red Hat Linux).

(continuare)

Tabelul 1-5 Continuare

Numele serviciului	Selectat prestabilit	Descriere
random	*	Serviciu necesar pentru generarea numerelor aleatoare cu un algoritm evoluat în sistemul dumneavoastră
routed		Actualizează tabelele de rutare IP utilizând protocolul RIP. Nu este necesar să activați acest serviciu decât dacă sistemul dumneavoastră va ruta pachete IP între mai multe rețele din cadrul organizației dumneavoastră
rusersd		Permite utilizatorilor de pe orice sistem din rețea să determine cine are sesiuni în sistem la un moment dat.
rwhod		Permite utilizatorilor aflați la distanță să obțină o listă a tuturor utilizatorilor de pe sistemul Linux care rulează serviciul rwhod.
sendmail	*	Deplasează mesaje de poștă între mașini diferite. Porniți serviciul dacă intenționați să trimiteți mesaje de pe sistem prin poșta electronică.
smb	*	Pornește și oprește serverele Samba smbd și nmbd, utilizate pentru a implementa serviciile LAN Manager pe un sistem Linux.
snmpd	*	Serviciul Simple Network Management Protocol (SNMP), utilizat în scopul gestionării rețelei.
sound	*	Salvează anumiți parametri ai plăcii de sunet la oprirea sistemului și îi reface la repornire.
syslog	*	Serviciu utilizat de multe alte programe (inclusiv de alte servicii) pentru a păstra un raport asupra diferitelor mesaje de stare sau de eroare într-un fișier (de regulă, acesta este fișierul /var/log/messages). Acest serviciu trebuie întotdeauna activat.
yplibd		Serviciu necesar pentru Network Information System (NIS). Nu trebuie să porniți ypbind decât dacă utilizați NIS.

Configurarea imprimantelor

După configurarea serviciilor, programul de instalare al sistemului Red Hat afișează o casetă de dialog (vezi figura 1-37), care vă întreabă dacă doriți să configurați o imprimantă. Dacă intenționați să tipăriți din sistemul Linux, selectați opțiunea Yes din această casetă de dialog.



Sugestie

Este necesar să selectați Yes din caseta de dialog din figura 1-37 chiar dacă pe moment nu aveți nici o imprimantă conectată direct la sistemul Linux. Trebuie să configurați o imprimantă chiar și în cazul în care intenționați să tipăriți la o imprimantă aflată la distanță. Acest pas vă permite să selectați o imprimantă partajată conectată la un alt sistem Linux sau la un sistem Windows 95.

După ce ați selectat Yes și ați apăsător Enter, programul de instalare vă cere să precizați tipul conexiunii cu imprimanta, după cum se observă în figura 1-38.

Puteți opta pentru unul din următoarele trei tipuri de conexiuni cu imprimanta:

- **Local:** Se referă la o imprimantă direct conectată la PC-ul pe care instalați Linux.

- **Remote lpd:** Se referă la o imprimantă conectată fizic la un alt sistem Linux din rețeaua locală.
- **LAN Manager:** Se referă la o imprimantă conectată la un alt PC din rețeaua locală și care rulează protocolul LAN Manager (utilizat de obicei pentru partajarea unor resurse, cum ar fi discurile sau imprimantele între PC-uri care rulează Windows 95 sau Windows NT).

Figura 1-37

Caseta de dialog care vă întreabă dacă doriți să configurați o imprimantă.

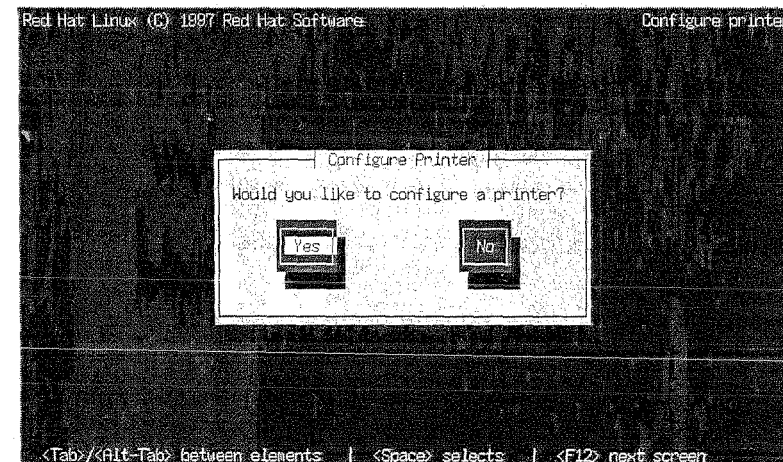
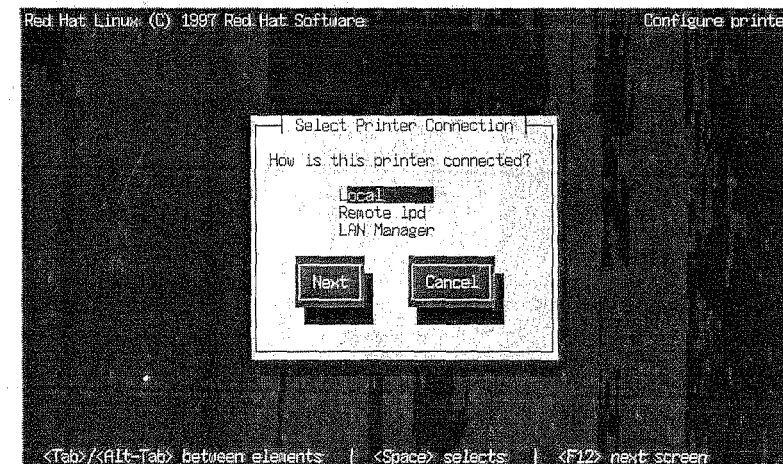


Figura 1-38

Selectarea tipului de conexiune cu imprimanta



Este necesar să specificați exact tipul conexiunii care se potrivește cu configurația dumneavoastră. După ce ați selectat un tip de conexiune, programul de instalare afișează o casetă de dialog cu două opțiuni standard, după cum se poate observa în figura 1-39.

Aceste opțiuni sunt denumite standard deoarece ele sunt necesare indiferent de tipul conexiunii cu imprimanta. Nu este nevoie să modificați aceste opțiuni standard; selectați pur și simplu butonul Next și apăsați Enter.

Următorul pas de configurare depinde de tipul conexiunii cu imprimanta pe care l-ați specificat în caseta de dialog din figura 1-38. Dacă ați optat pentru o imprimantă locală, programul de

instalare va afișa o casetă de dialog (vezi figura 1-40), care vă cere să precizați portul la care este conectată fizic imprimanta.

Figura 1-39
Selectarea
opțiunilor
standard pentru
imprimantă

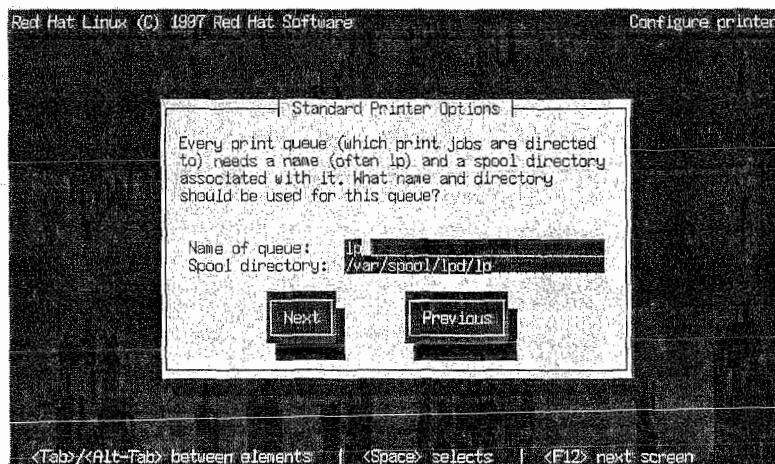
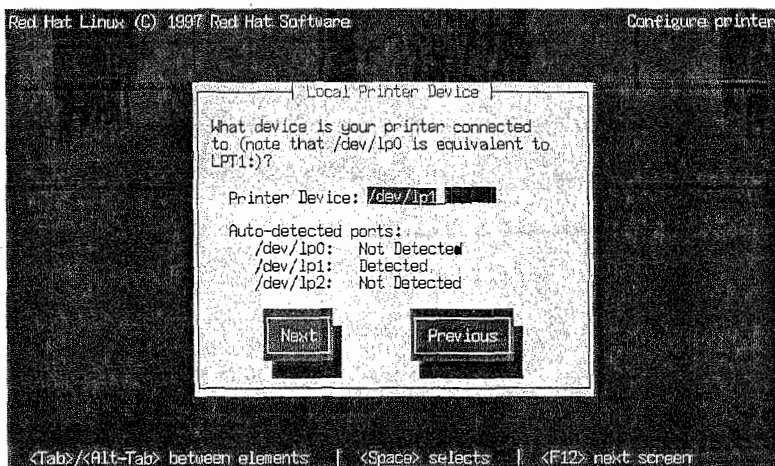


Figura 1-40
Alegerea portului
pentru o
imprimantă locală



De regulă, imprimanta este conectată la portul paralel al PC-ului. În Windows, porturile paralele sunt denumite LPT1, LPT2 etc. În Linux, numele dispozitivelor corespunzătoare sunt /dev/lp0, /dev/lp1 etc. După cum se observă și în caseta de dialog din figura 1-40, programul de instalare vă va comunica porturile pe care le-a detectat și va completa într-o zonă de text numele celui mai plauzibil port. Este necesar să verificați că această informație este corectă (și dacă este cazul, să introduceți numele portului corect); apăsați apoi Enter pentru a continua configurarea imprimantei.

Următoarea casetă de dialog (figura 1-41) vă cere să identificați tipul imprimantei dumneavoastră într-o listă de imprimante cunoscute. Selectați tipul imprimantei și apoi apăsați Enter pentru a continua operația de configurare.

Programul de instalare va afișa în continuare o altă casetă de dialog (figura 1-42) care vă cere să selectați dimensiunea foi de hârtie și rezoluția imprimantei (care de regulă este de 600x600 puncte pe inch pentru majoritatea imprimantelor laser moderne). Trebuie să selectați valori corecte pentru acești parametri.

Figura 1-41
Selectarea tipului
imprimantei

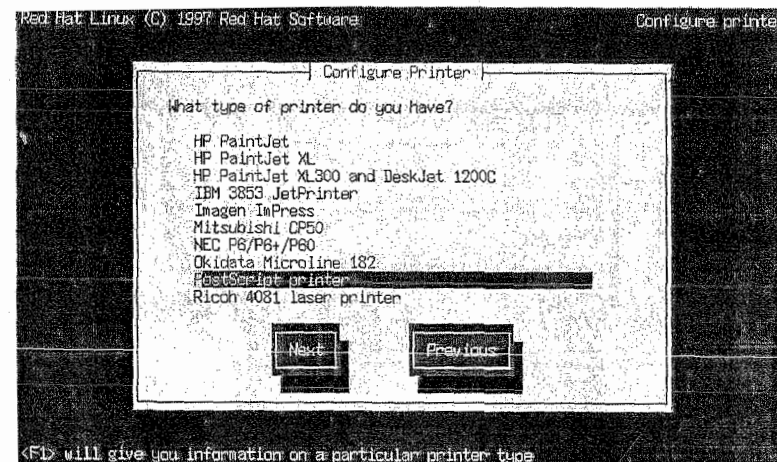
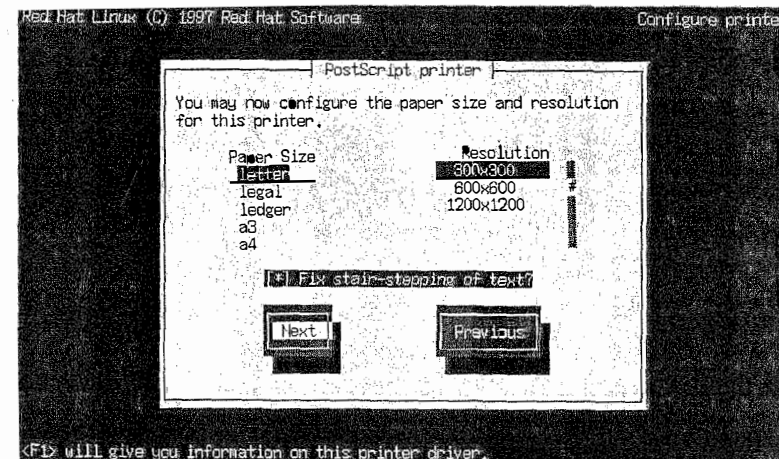


Figura 1-42
Casetă de dialog
prin care puteți
selecta
dimensiunea foi
de hârtie și
rezoluția
imprimantei



Un alt element din caseta de dialog din figura 1-42 este o opțiune denumită Fix stair-stepping of text. Aceasta se referă la o problemă care apare atunci când se imprimă fișiere text în Linux – deoarece în aceste fișiere fiecare linie se termină cu un singur caracter Line Feed, în timp ce în DOS sau Windows liniile se termină cu o secvență de două caractere, Carriage Return urmat de Line Feed. Imprimanta reacționează la aceste caractere într-o manieră similară vechilor mașini de scris – caracterul Line Feed determină avansul la o linie nouă, în timp ce Carriage Return determină deplasarea la începutul liniei. Atunci când tipăriți un fișier text DOS, liniile vor fi tipărite în mod evident una după alta. Când însă

imprimanta tipărește un fișier text Linux, în care liniile se termină doar cu caracterul Line Feed, aspectul textului va fi astfel:

Linia 1 se termina aici

Linia 2 aici

si asa mai departe

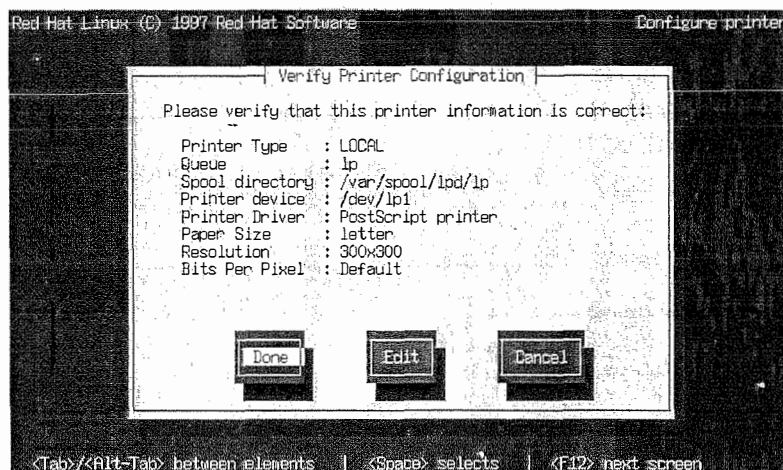
in forma de scara.

Pentru a evita acest efect de scară, denumit și „salt în scară” (stair-stepping), este suficient să marcați caseta numită Fix stair-stepping of text în caseta de dialog din figura 1-42, iar sistemul Linux va rezolva această problemă, adăugând singur caracterul Carriage Return înaintea fiecărui Line Feed ori de câte ori tipăriți fișiere text.

După ce ați selectat opțiunile din figura 1-42 și ați apăsă Enter, programul de instalare vă prezintă într-o altă casetă toate opțiunile (figura 1-43), astfel încât să puteți verifica întreaga configurație a imprimantei.

Figura 1-43

Verificarea parametrilor imprimantei



După ce ați confirmat că acești parametri sunt cei corecți, apăsați Enter pentru a trece la următorul pas din procesul de configurare.

Stabilirea parolei de root

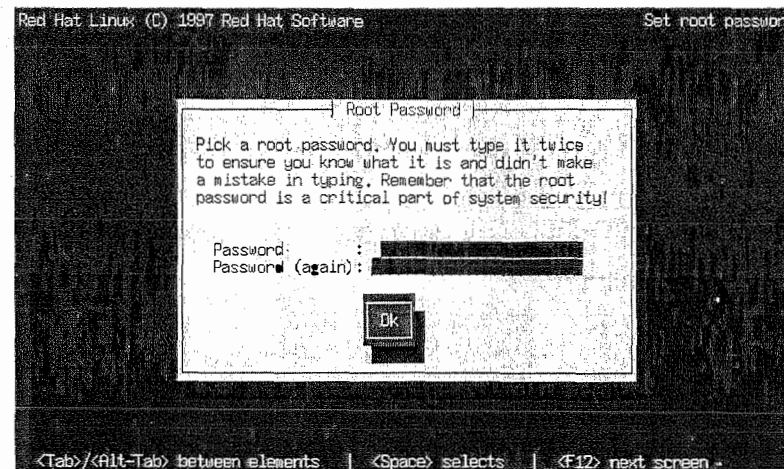
După ce ați terminat de configurat imprimanta, programul de instalare vă cere să introduceți o parolă de root, după cum se observă în figura 1-44.

Utilizatorul root este superutilizator în sistemul Linux. Din cauză că superutilizatorul are drepturi depline în sistem, trebuie să alegeți o parolă pe care să v-o amintiți, dar pe care ceilalți să nu o poată întui cu ușurință. De regulă, este bine ca parola să aibă lungimea de cel puțin opt caractere, să conțină atât litere cât și cifre; pentru siguranță, este recomandat să introduceți și un caracter special, cum ar fi + sau *.

Introduceți parola pe prima linie și apoi reintroduceți-o pe următoarea. Parola nu apare în clar pe ecran. Este necesar să introduceți parola de două ori și ambele valori introduse trebuie să coincidă înainte ca programul de instalare să accepte parola. După ce ați tastat de două ori parola, selectați OK și apăsați Enter pentru a trece la următorul pas de configurare.

Figura 1-44

Caseta de dialog care permite alegerea unei parole de root



Crearea unei dischete de pornire

După ce ați stabilit parola de root, programul de instalare afișează o casetă de dialog care vă întreabă dacă doriți să creați o dischetă de pornire. Această dischetă poate fi utilizată pentru pornirea sistemului în cazul în care nucleul Linux de pe disc este corupt sau dacă programul Linux Loader (care va fi prezentat în secțiunea imediat următoare) nu funcționează. Apăsați Enter pentru a răspunde cu Yes. Programul de instalare vă cere să introduceți o dischetă formatată în unitatea A a PC-ului. Introduceți discheta în unitatea A și apoi apăsați Enter (atenție, toate informațiile aflate pe dischetă se vor pierde).

Programul de instalare copiază nucleul Linux și alte câteva fișiere pe dischetă. După ce programul de inițializare pregătește discheta de pornire, scoateți discheta din unitatea A, etichetați-o în mod corespunzător și păstrați-o pentru a fi disponibilă în viitor.

Puteți utiliza această dischetă pentru a porni sistemul Linux pe PC-ul dumneavoastră. În secțiunea următoare, veți învăța cum să instalați programul Linux Loader pe unitatea C. Acest proces implică însă alterarea unei părți critice a discului. Dacă nu doriți să alterați discul sau dacă programul Linux Loader nu funcționează dintr-un alt motiv, puteți porni sistemul Linux de pe discheta de care tocmai ați creat-o.

Instalarea programului LILO

LILO reprezintă prescurtarea denumirii Linux Loader – un program aflat pe hard-disc și care permite încărcarea sistemului Linux de pe disc. Dacă pe disc mai aveți instalat unul din sistemele Windows sau OS/2, puteți de asemenea să configurați LILO pentru a încărca oricare dintre aceste sisteme de operare.

În acest ultim pas de configurare, programul de instalare a sistemului Red Hat afișează o casetă de dialog (figura 1-45) care vă întreabă unde doriți să instalați programul LILO, cunoscut și sub numele de *bootloader* (încărcător la pornire).

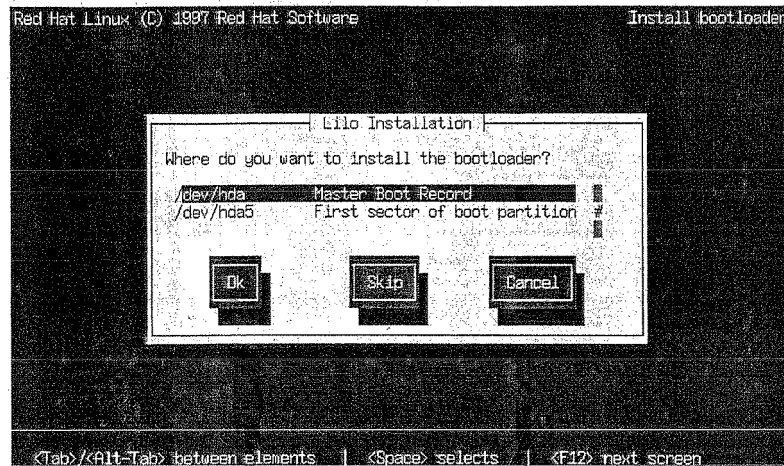
Caseta de dialog vă oferă posibilitatea de a instala LILO în unul din următoarele două locuri:

- Înregistrarea principală de boot (Master Boot Record, prescurtat MBR), care se găsește în primul sector al discului PC-ului dumneavoastră (adică al unității C)

■ Primul sector al partiției în care ați încărcat sistemul Linux

Figura 1-45

Precizarea locului în care va fi instalat programul LILO



Selectați unul din aceste două locuri – de preferință înregistrarea principală de boot – apăsați apoi tasta Tab pentru a selecta opțiunea OK și apoi apăsați **Enter**.

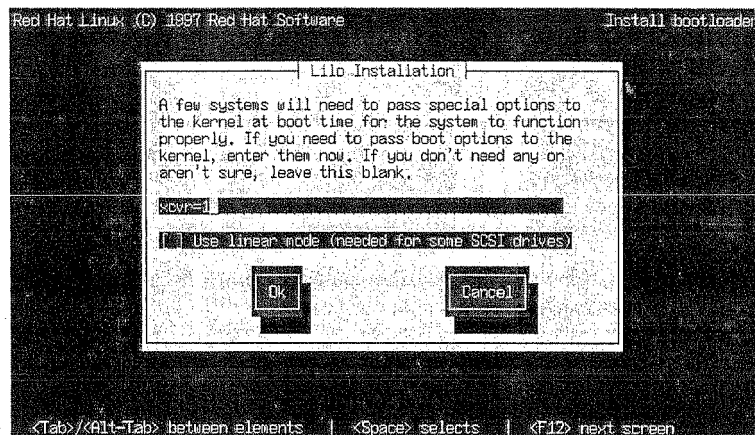


Atenție!

Deși puteți evita instalarea programului LILO cu ajutorul butonului Skip, este necesar să instalați programul LILO, în caz contrar fiind în situația de a nu putea porni sistemul Linux după ce terminați instalarea. În același timp, instalarea programului LILO în înregistrarea principală de boot poate fi riscantă. Dacă LILO nu este configurat corect, PC-ul nu va mai putea porni de pe hard-disc. Din nefericire, programul de instalare a sistemului Red Hat nu oferă o soluție alternativă pentru pornirea sistemului Linux. Trebuie însă să așteptați până la prima pornire a sistemului Linux înainte de a pregăti discheta de pornire pentru a porni sistemul de pe unitatea A a PC-ului dumneavoastră. După ce ați selectat locul în care va fi instalat LILO, programul de instalare va afișa o altă casetă de dialog (figura 1-46) care vă permite să introduceți parametri speciali de care sistemul Linux poate avea nevoie la pornire.

Figura 1-46

Opțiuni de pornire pentru sistemul Linux

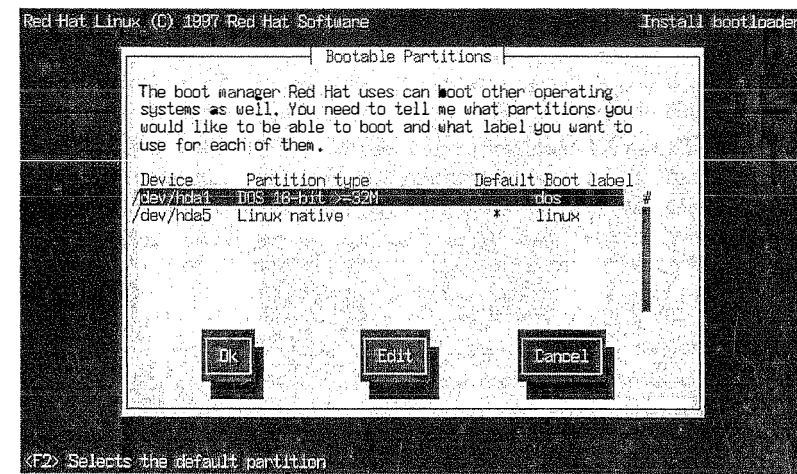


În funcție de configurația hardware a sistemului dumneavoastră, aveți sau nu nevoie de opțiuni speciale. De exemplu, să considerăm un sistem cu o placă de rețea 3Com 3C503, conectată la rețeaua Ethernet locală cu ajutorul unui transceiver extern. Atunci când sistemul Linux încarcă driverul pentru placa 3Com 3C503, este necesar să specificăm opțiunea `xvr=1` pentru a ne asigura că driverul utilizează transceiverul extern. Această opțiune trebuie introdusă pe prima linie a casetei de dialog, după cum se observă în figura 1-46.

După ce ați introdus toate opțiunile necesare, selectați butonul OK și apăsați **Enter** pentru a trece la instalarea programului LILO. Programul de instalare a sistemului Red Hat va afișa o altă casetă de dialog (figura 1-47), care vă permite să marcați și alte partiții ca fiind de boot.

Figura 1-47

Marcarea altor partiții de boot sub LILO



Caseta de dialog vă prezintă un tabel care cuprinde partiția Linux și orice alte partiții care pot conține alte sisteme de operare. De regulă, tabelul conține două elemente: partiția Linux și partiția DOS (pe care în cele mai multe cazuri este instalat sistemul Windows 95). Fiecare element din tabel reprezintă un sistem de operare care poate fi pornit cu ajutorul programului LILO.

După ce este instalat, la fiecare pornire a PC-ului de pe discul fix, LILO rulează și afișează un prompt. La acest prompt, puteți introduce numele sistemului de operare care va fi pornit – ultima coloană a tabelului din figura 1-47 conține numele pe care le puteți introduce la promptul LILO. În acest caz, puteți tasta `linux` pentru a porni sistemul Linux, sau `dos` pentru a porni de pe partiția DOS (încărcând sistemul Windows 95, dacă acesta a fost instalat pe partiția DOS).

Dacă nu introduceți nimic la promptul LILO, acesta așteaptă câteva secunde și încarcă sistemul de operare prestabilit. Sistemul de operare prestabilit este cel marcat cu un asterisc în coloana Default din figura 1-47 (în cazul de față, acesta este Linux).

Dacă doriți să modificați aceste valori, selectați opțiunea Edit și apăsați apoi **Enter**. Dacă acceptați valorile selectate în figura 1-47, apăsați tasta Tab pentru a selecta OK și apoi apăsați **Enter**. Programul LILO va fi instalat, ceea ce va încheia instalarea sistemului Red Hat Linux.

Pornirea sistemului Linux pentru prima dată

După ce ați instalat LILO, programul de instalare a sistemului Red Hat Linux va reporni automat sistemul. PC-ul va parcurge sevența obișnuită de pornire, va încărca LILO de pe unitatea C, ceea ce va determina apariția promptului:

```
LILO boot:
```

Dacă la instalarea LILO ați specificat partiția Linux ca fiind cea prestabilită, este suficient să așteptați: după câteva secunde, LILO va porni sistemul Linux.

Dacă doriți să porniți de pe o altă partiție (de exemplu de pe partiția DOS), apăsați tasta Tab. LILO va afișa numele tuturor partițiilor de boot disponibile. De exemplu, pe ecran poate apărea secvența:

```
linux dos
```

Puteți apoi să tastați numele partiției de pe care doriți să porniți (sau să apăsați Enter pentru a porni de pe prima partiție).

După ce LILO pornește sistemul Linux, va apărea o listă lungă de mesaje introductive, care includ numele dispozitivelor fizice detectate de sistemul Linux. Unul dintre primele mesaje este Calibrating delay loop... ok - 299.01 BogoMIPS; numărul care precede BogoMIPS depinde de tipul procesorului din configurație. *BogoMIPS* reprezintă un termen din jargonul Linux care este subiectul a nenumărate discuții în diferitele grupuri USENET dedicate sistemului Linux. (Usenet este o colecție întinsă de calculatoare care schimbă între ele mesaje prin poșta electronică și știri. Mai multe despre Usenet veți afla în capitolele 18 și 19.) Sistemul Linux utilizează valoarea măsurată în BogoMIPS în cazurile în care sistemul de operare trebuie să aștepte pentru o anumită perioadă.

Ce reprezintă BogoMIPS ?

La pornirea sistemului Linux, observați un mesaj de genul Calibrating delay loop... ok - 299.01 BogoMIPS, care are un număr prezent înaintea termenului BogoMIPS. Acest termen este unul dintre aceia care pot induce în eroare pe novici. Această notă este destinată explicării semnificației acestui termen.

După cum poate știți, MIPS reprezintă abrevierea de la *Millions of Instructions Per Second* (milioane de instrucțiuni pe secundă) - o măsură a vitezei cu care un calculator poate rula un program. (De altfel, MIPS nu reprezintă o unitate adecvată de măsură, deoarece compararea vitezei în MIPS pentru două tipuri de calculatoare diferite este dificilă.) BogoMIPS este format prin compunere de la *bogus* (fals) și MIPS, și se referă la indicația asupra vitezei calculatorului. Sistemul Linux utilizează numărul BogoMIPS pentru calibrarea unei bucle de întârziere, în care calculatorul repetă încontinuu un

set de instrucțiuni până la trecerea unei anumite durate.

Numerele BogoMIPS sunt cuprinse între 1 și 300 sau chiar mai mult, în funcție de tipul procesorului (386, 486 sau Pentium). Un sistem 80386 DX la 33 MHz are un număr BogoMIPS de aproximativ 6, în timp ce pentru un 80486DX2/66 cu viteză de 66 MHz acest număr este de circa 33. Valoarea BogoMIPS pentru sistemele cu procesoare Pentium mai vechi este sensibil egală (sau chiar mai mică) față de valoarea pentru procesoarele 80486, deoarece calculul acestui număr nu utilizează anumite facilități speciale (cum ar fi capacitatea de a executa anumite instrucțiuni în paralel) ale procesoarelor Pentium. Pe un calculator Pentium la frecvența de 75 MHz, sistemul Linux raportează o valoare BogoMIPS de 30,22. Pe un calculator mai recent, Pentium MMX la 200 MHz, numărul BogoMIPS este 299,01.

La sfârșitul acestor mesaje, va apărea promptul Linux, care arată astfel:

```
Red Hat Linux release 5.1 (Mannhattan)
Kernel 2.0.34 on an i586
hostname login:
```

unde *hostname* este numele pe care l-ați dat sistemului atunci când ați configurat rețeaua. Dacă nu ați configurat rețeaua, numele sistemului este localhost.

Cum până în acest moment sistemul nu are nici un alt utilizator, tastați **root** și apoi apăsați Enter. Introduceți apoi parola de root (cea pe care ați stabilit-o pe parcursul instalării) pentru a intra în sistem ca superutilizator. Puteți efectua acum câteva activități esențiale, învățând totodată cum să opriți sistemul Linux.

Ce facem dacă am uitat parola de root?

Dacă ați uitat parola de root, executați următorii pași pentru a stabili o nouă parolă:

1. Porniți PC-ul ca de obicei. La promptul LILO, introduceți numele partiției Linux urmat de cuvântul single, după cum urmează (textul pe care trebuie să-l introduceți este tipărit îngroșat):

```
LILO boot: linux single
```

În acest mod, sistemul Linux va porni în mod obișnuit, dar va rula în mod single-user, mod care nu necesită introducerea unei parole. După pornirea sistemului, va apărea imediat următorul prompt:

```
bash#
```

2. Utilizați comanda **passwd** pentru a modifica parola de root, după cum urmează:

```
bash# passwd
New UNIX password:
```

Introduceți parola pe care doriți să o utilizați (aceasta nu va apărea scrisă pe ecran), după care apăsați Enter. Sistemul Linux vă cere să introduceți parola încă o dată:

```
Retype new UNIX password:
```

Introduceți parola încă o dată și apăsați Enter. Dacă veți introduce aceeași parolă în ambele cazuri, comanda **passwd** va modifica parola și apoi va afișa mesajul:

```
passwd: all authentication tokens updated successfully
```

(adică **passwd**: Toate elementele necesare pentru autentificare au fost modificate cu succes)

3. Reporniți acum PC-ul apăsând combinația de taste Ctrl+Alt+Delete. La pornire, sistemul Linux afișează din nou promptul obișnuit de intrare în sistem. Acum trebuie să puteți intra în sistem ca root utilizând noua parolă.

Crearea unei dischete de pornire

După ce ați pornit cu succes sistemul Linux și ați intrat ca root, trebuie ca imediat să creați o dischetă de pornire a sistemului Linux. Veți putea utiliza această dischetă pentru a porni sistemul Linux în cazul în care nucleul Linux de pe hard-disc este corupt. Această dischetă vă este necesară înainte de a efectua orice modificare a nucleului Linux, cum ar fi o actualizare cu o versiune mai nouă, urmând indicațiile care vor fi prezentate în capitolul 2.

Pentru a crea discheta de pornire, procedați după cum urmează:

1. Introduceți o dischetă formatată în unitatea A a PC-ului. Dacă dispuneți de o dischetă neformatată, o puteți formata utilizând comanda:

```
fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 KB.
Formatting ... done
Verifying ... done
```

2. Copiați fișierul `vmlinux` din directorul `/boot` pe dischetă. (Fișierul `vmlinux` este chiar nucleul sistemului Linux.) Utilizați comanda:

```
cd /boot
cp vmlinux /dev/fd0
cp: overwrite '/dev/fd0' ? Y
```

3. Determinați dispozitivul rădăcină al nucleului – adică partiția unde se găsește rădăcina sistemului de fișiere – utilizând comanda:

```
rdev
/dev/hda3/
```

În cazul de mai sus, dispozitivul rădăcină este `/dev/hda3`; numele acestuia poate fi diferit pe sistemul dumneavoastră.

4. Stabiliți dispozitivul rădăcină al nucleului de pe dischetă. Utilizați același dispozitiv pe care comanda `rdev` l-a detectat în pasul precedent. De exemplu, pentru ca dispozitivul rădăcină să fie `/dev/hda3`, vom introduce:

```
rdev /dev/fd0 /dev/hda3
```

5. Marcați dispozitivul rădăcină ca fiind disponibil numai pentru citire. Pentru aceasta, utilizați comanda:

```
rdev -R /dev/fd0 1
```

În acest fel, sistemul Linux va monta inițial rădăcina sistemului de fișiere doar cu drept de citire. Sistemul Linux verifică prezența unor erori în sistemul de fișiere și, dacă nu găsește nici o eroare, montează sistemul de fișiere cu drepturi de citire și scriere. Prin montarea inițială a dispozitivului rădăcină numai cu drept de citire, puteți evita câteva mesaje de avertisment și de eroare.

După ce ați pregătit discheta, este bine să o verificați. Parcurgeți următoarele patru secțiuni. Apoi, când sunteți gata de a opri sistemul, introduceți discheta în unitatea A și tastați comanda:

```
/sbin/shutdown -r now
```

După ce PC-ul repornește, sistemul Linux va fi încărcat în mod normal și va apărea un prompt de intrare în sistem. Dacă acesta nu apare, scoateți discheta din unitatea A și apăsați **Ctrl+Alt+Delete** pentru a porni de pe discul fix. Puteți apoi repeta procedura de creare a unei dischete de pornire.

Pornirea sistemului X

În unul din pașii din secvența de instalare a sistemului Red Hat Linux, ați configurat sistemul X Window. În urma aceluși pas de configurare, s-a creat un fișier care conține informații referitoare la mouse-ul, tastatura, placa video și monitorul care se găsesc în configurația dumneavoastră. (Veți afla totul despre fișierul de configurare a sistemului X în capitolul 4.) Acum, porniți imediat sistemul X și testați dacă fișierul de configurare funcționează. Pentru a porni sistemul X, introduceți la promptul Linux următoarea comandă:

```
startx
```

Veți observa derularea pe ecran a unui număr de mesaje în timp ce serverul X pornește.



Atenție!

Dacă ecranul capătă un aspect ciudat, apăsați combinația **Ctrl+Alt+Backspace** pentru a opri serverul X. Dacă aceasta nu oprește serverul X, apăsați butonul **Reset** al calculatorului. În caz contrar, se poate produce defectarea monitorului.

Dacă fișierul de configurare nu funcționează, vor apărea câteva mesaje de eroare: unul dintre acestea spune că serverul X nu poate fi pornit, acesta fiind urmat de altul, în care un proces raportează că nu se poate conecta la serverul X:

```
_X11TransSocketUNIXConnect: Can't connect: errno=111
giving up.
xinit: Connection refused (errno 111): unable to connect to X server
xinit: No such process (errno 3): Server error.
```



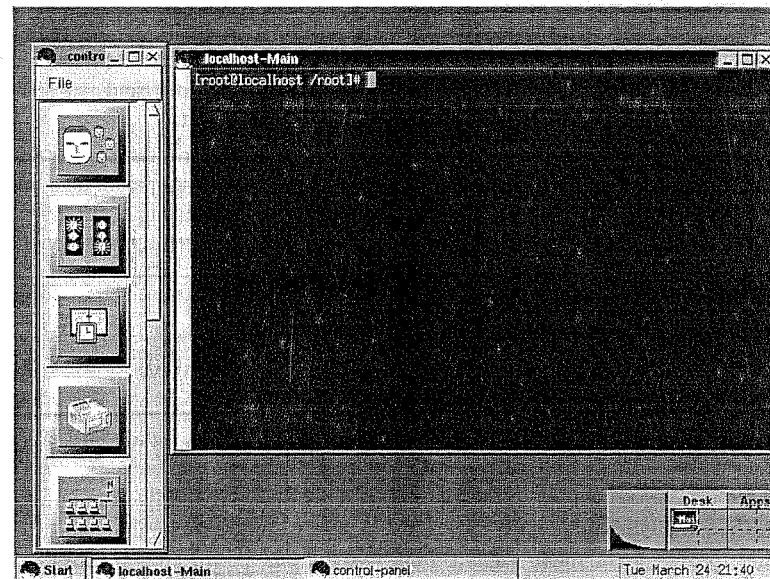
Referință încrucișată

Deși se pare că mesajele de eroare se referă la socket-uri și la conexiuni, motivul pentru care acestea apar este legat de obicei de fișierul de configurare – serverul X nu a putut porni din cauza unor probleme cu unii dintre parametrii din fișierul de configurare. O problemă relativ frecventă este aceea că serverul X nu a putut detecta un mod video potrivit (adică o rezoluție a ecranului) pentru monitorul și placa video din configurație. Consultați capitolul 10 pentru a afla mai multe despre fișierul de configurare a serverului X și despre parametrii pe care îi puteți ajusta pentru a putea rula X pe sistemul dumneavoastră.

Dacă totul este în regulă, culoarea ecranului va deveni gri pentru un timp și veți observa un cursor de forma literei X. În scurt timp, culoarea de fond a ecranului se modifică și apar câteva ferestre, după cum se poate observa și în figura 1-48.

Figura 1-48

Prima pornire a serverului X



Referință încrucișată

În funcție de placa video și monitorul din configurație, aspectul ecranului dumneavoastră poate fi diferit față de cel prezentat în figura 1-48. Mai precis, ecranul poate reprezenta doar o mică parte a unui ecran virtual mai mare; în acest caz, trebuie să deplasați cursorul mouse-ului în jos pentru a vedea bara cu

instrumente. În acest moment, puteți ignora total aceste nuanțe. Capitolul 5 prezintă structura diferitelor elemente de pe ecran și modul cum puteți realiza personalizarea acestora.

Priviți ecranul inițial al sistemului X Window din figura 1-48. Veți observa trei elemente de bază pe acest ecran:

- De-a lungul marginii din stânga, se observă un panou de control – o fereastră cu un număr de butoane mari, fiecare cu câte o anumită imagine pe suprafață. Aceste pictograme reprezintă instrumente (programe) pe care le puteți lansa executând clic pe pictograma corespunzătoare. Capitolul 6 vă prezintă modul în care puteți utiliza câteva instrumente din panoul de control pentru a efectua câteva sarcini de administrare a sistemului Linux.
- La dreapta panoului de control – ocupând majoritatea spațiului de pe ecran – se găsește o fereastră terminal. În această fereastră, puteți introduce comenzi Linux. Capitolul 6 prezintă câteva comenzi Linux, iar anexa B cuprinde o listă a celor mai des utilizate comenzi Linux.
- O bară de taskuri, similară cu cea din Windows 95, apare de-a lungul marginii de jos a ecranului. Ca și în Windows 95, bara de taskuri are un buton Start și câte un buton pentru fiecare din ferestrele deschise. La marginea din dreapta a barei de taskuri, sunt afișate data și ora curentă.

Adăugarea conturilor pentru utilizatori

Din moment ce interfața grafică X a fost configurată și funcționează, puteți efectua cu ajutorul acesteia una din activitățile esențiale în administrarea sistemului – adăugarea unor noi conturi pentru utilizatori în sistem.



Sugestie

Este o idee bună să creați cel puțin un alt cont de utilizator pe lângă cel de root. Chiar dacă sunteți singurul utilizator al sistemului, utilizarea sistemului cu mai puține privilegii este recomandată, deoarece în acest mod nu puteți distruge din neatenție anumite fișiere de sistem esențiale. Atunci când este necesar, puteți intra ca root pentru a efectua sarcinile de administrare a sistemului.

Ca o regulă, sistemul Red Hat Linux include mai multe instrumente grafice de administrare a sistemului Linux, care apar sub forma unor pictograme într-un panou de control (fereastra înaltă din partea stângă a ecranului din figura 1-48). Dacă deplasați mouse-ul deasupra fiecărei pictograme, va apărea un mesaj al cărui rol este de a vă prezenta numele instrumentului respectiv. Prima pictogramă, cea care prezintă imaginea unui chip omenesc, este destinată adăugării conturilor pentru utilizatori.

Pentru a adăuga un nou cont, vom proceda după cum urmează:

1. Se execută clic pe prima pictogramă din panoul de control. Va apărea fereastra programului User Configurator, prezentată în figura 1-49.
2. Executați clic pe butonul Add din fereastra principală din figura 1-49. Va apărea caseta de dialog Edit User Definition, prezentată în figura 1-50.
3. Introduceți informațiile necesare. Aceasta înseamnă că trebuie să introduceți numele utilizatorului, parola și directorul personal de lucru. Pentru contul personal, autorul a ales directorul de lucru /home/naba. Dumneavoastră puteți utiliza un director similar pentru contul propriu. După ce ați completat câmpurile, executați clic pe butonul Done. Vă veți întoarce astfel la fereastra principală din programul User Configurator (cea din figura 1-49).
4. Executați clic pe butonul Quit. Va apărea o casetă de dialog, care vă cere o confirmare. Executați clic pe butonul Save and Quit pentru a salva informațiile asociate noului utilizator creat și a părăsi ulterior programul.

Figura 1-49

Fereastra principală a programului User Configurator

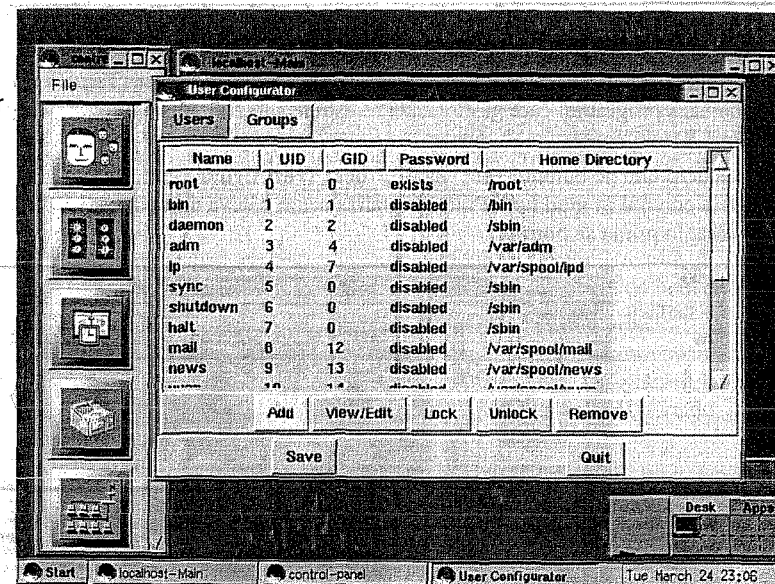
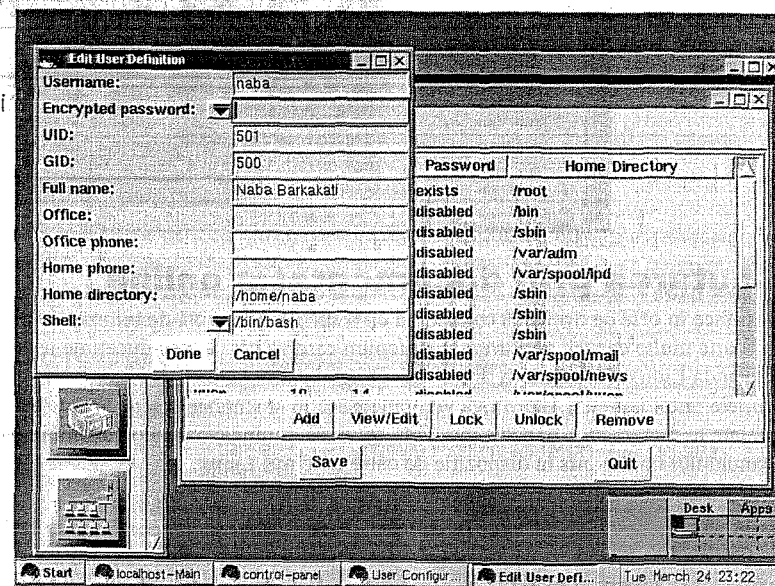


Figura 1-50

Introducerea informațiilor pentru contul unui nou utilizator



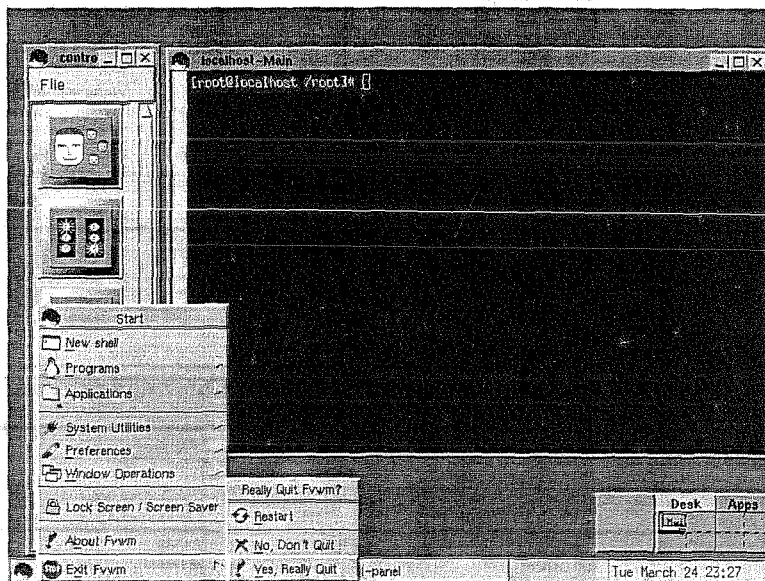
Părăsirea mediului X

După ce ați încercat serverul X și ați utilizat unul dintre instrumentele grafice din sistemul X, este rândul să învățați cum se poate părăsi mediul X. Pentru a opri sistemul X Window, urmați pașii descriși în continuare:

1. Executați clic pe butonul Start din bara de taskuri (în colțul din stânga-jos al ecranului). Va apărea meniul Start (similar cu cel din Windows 95).
2. Deplasați cursorul mouse-ului deasupra elementului Exit Fvwm (Termenul Fvwm desemnează programul care gestionează ferestrele sub X – acesta trebuie părăsit pentru a opri serverul X). Va apărea un alt meniu.
3. Executați clic pe butonul Yes, Really Quit (vezi figura 1-51). Ecranul X va dispărea și veți reveni la ecranul în mod text. Puteți vizualiza acum câteva dintre mesajele pe care serverul X le-a afișat la prima sa pornire.

Figura 1-51

Oprirea sistemului
X Window



Căutarea prin documentația online

Veți face în cele ce urmează cunoștință cu o sursă importantă de informație din sistemul Linux. În foarte multe cazuri, veți urma instrucțiuni care vă cer să introduceți de la tastatură o comandă Linux. După câțva timp, veți ajunge în situația în care vă veți aminti cât de cât numele unor comenzi, dar nu vă veți putea aminti și sintaxa exactă cu care acestea trebuie apelate în fiecare caz. În acest moment, salvarea dumneavoastră o constituie paginile manualului online pus la dispoziție de către sistemul Linux.

Puteți vizualiza pagina de manual – denumită în mod uzual *pagina man* – asociată unei anumite comenzi utilizând comanda *man*. (Este necesar să vă reamintiți de această comandă pentru a putea beneficia de avantajele sistemului de help online.) Pentru a vizualiza de exemplu pagina *man* pentru comanda *passwd*, tastați următoarea comandă:

```
man passwd
```

Sistemul Linux afișează informația din paginile *man* pagină cu pagină. Apăsăți tasta *Spacebar* pentru a vă deplasa la pagina următoare. Când ați terminat, apăsați tasta *q* pentru a vă întoarce la promptul sistemului Linux.

După ce am scos în evidență utilitatea paginilor de help online, trebuie să amintim că termenul de comandă Linux se referă la orice fișier executabil, de la un script care conține apeluri ale altor comenzi Linux și până la programele executabile standard din sistemul Linux. Deși există pagini *man* pentru majoritatea programelor standard, multe programe nu dispun de help online. Spre exemplu, dacă tastați *help* la promptul sistemului Linux, obțineți lista comenzilor care sunt definite intern în cadrul procesorului de comenzi (denumit *shell* în UNIX). Dacă însă tastați comanda *man help*, sistemul Linux dă următorul răspuns:

```
No manual entry for help
```

Cu toate acestea, atunci când nu vă mai amintiți prea bine o anumită comandă, merită să utilizați comanda *man* pentru a vedea dacă acea comandă dispune de un sistem de help online.

O altă formă de documentație online o constituie fișierele HOWTO din directorul */usr/doc/HOWTO* al sistemului (aceste fișiere vor fi instalate dacă selectați și componenta Extra Documentation în procesul de instalare.) Fișierele HOWTO au toate în comun sufixul *-HOWTO.gz*. Fiecare dintre aceste fișiere este un fișier text, care a fost compactat utilizând comanda *gzip*.

Fiecare fișier HOWTO conține informații despre anumite domenii din sistemul Linux, cum ar fi dispozitivele fizice pe care acesta le suportă sau modul de creare a unei dischete de pornire. Tabelul 1-6 prezintă lista fișierelor HOWTO care trebuie să se găsească în directorul */usr/doc/HOWTO* al sistemului Linux, în cazul în care ați instalat și componenta Extra Documentation.

Pentru vizualizarea oricăruia dintre aceste fișiere, deplasați-vă în directorul */usr/doc/HOWTO* și utilizați comanda *zcat* după cum urmează:

```
cd /usr/doc/HOWTO
zcat nume-subiect-HOWTO.gz | more
```

unde *nume-subiect* este prima parte a numelui fișierului, din care am exclus sufixul comun *-HOWTO.gz*. Prin urmare, pentru a vizualiza fișierul *Bootdisk-HOWTO.gz*, tastați:

```
zcat Bootdisk-HOWTO.gz | more
```



Sugestie

Nu este nevoie să introduceți întreg numele fișierului. Tastați doar *zcat Bootd* și apoi apăsați tasta *Tab*. Procesorul de comenzi din Linux – *shell* – va completa în mod automat numele fișierului care se potrivește cu ceea ce ați tastat. Tastați apoi *| more* pentru a completa linia de comandă.

Tabelul 1-6 Fișierele Linux HOWTO din directorul */usr/doc/HOWTO*

Numele fișierului HOWTO	Conținutul
3Dfx-HOWTO.gz	Modul în care se poate configura sistemul Linux pentru a suporta acceleratorul grafic 3Dfx.
AX25-HOWTO.gz	Modul de instalare și de configurare a sistemului Linux pentru a suporta protocolul AX.25, utilizat de operatorii radioamatori din întreaga lume.
Access-HOWTO.gz	Modul în care sistemul Linux poate deveni utilizabil de către persoanele cu handicap.
Alpha-HOWTO.gz	Scurtă descriere a arhitecturii procesorului RISC pe 64 de biți Alpha, produs de firma Digital Equipment Corporation (sistemul Linux poate rula și pe procesoare Alpha).

(continuare)

Tabelul 1-6 Continuare

Numele fișierului HOWTO	Conținutul
Assembly-HOWTO.gz	Modul de programare în limbaj de asamblare în sistemul Linux.
Benchmarking-HOWTO.gz	Cum putem măsura performanțele unui sistem Linux – măsurarea vitezei cu care sistemul efectuează anumite activități.
BootPrompt-HOWTO.gz	Lista parametrilor care pot fi transmiși sistemului Linux la pornire (la promptul LILO boot:)
Bootdisk-HOWTO.gz	Cum putem crea discheta de pornire, discheta suplimentară și alte dischete utile pentru sistemul Linux.
CD-Writing-HOWTO.gz	Cum putem înregistra un CD-ROM utilizând un CD Recorder instalat în sistemul Linux.
CDROM-HOWTO.gz	Cum putem instala, configura și utiliza unitățile CD-ROM sub Linux.
Chinese-HOWTO.gz	Cum putem configura sistemul Linux pentru utilizarea setului chinezesc de caractere și versiunii chinezești a sistemului X Window.
Commercial-HOWTO.gz	O listă a programelor comerciale disponibile pentru sistemul Linux.
Consultants-HOWTO.gz	O listă a persoanelor fizice sau firmelor care oferă consultanță în ceea ce privește sistemul Linux.
Cyrillic-HOWTO.gz	Cum putem formata și tipări documente în limba rusă într-un sistem Linux.
DNS-HOWTO.gz	Cum putem configura serverul de nume (DNS) într-un sistem Linux.
DOS-to-Linux-HOWTO.gz	Cum ne putem folosi cunoștințele de MS-DOS în Linux.
DOSEMU-HOWTO.gz	Cum putem configura și utiliza emulatorul MS-DOS DOSEMU.
Danish-HOWTO.gz	Cum putem configura sistemul Linux pentru a utiliza setul danez de caractere.
Database-HOWTO.gz	Cum putem configura sistemul de baze de date relaționale PostgreSQL în Linux (PostgreSQL este distribuit împreună cu sistemul Red Hat Linux pe CD-ROM-ul atașat.).
Disk-HOWTO.gz	Cum putem utiliza cât mai eficient mai multe discuri și mai multe partiții într-un sistem Linux.
Distribution-HOWTO.gz	O listă a distribuțiilor sistemului Linux, cu precădere a celor comerciale livrate pe CD-ROM-uri.
ELF-HOWTO.gz	Cum putem compila și rula programe în formatul binar ELF (Executable and Linking Format); versiunea sistemului Linux de pe CD-ROM-ul atașat este deja configurată pentru a putea rula programe binare în format ELF.
Emacspeak-HOWTO.gz	Cum putem oferi ajutor utilizatorilor nevăzători utilizând Emacspeak, un subsistem al editorului Emacs care oferă răspunsuri sub formă de voce sintetizată.
Ethernet-HOWTO.gz	Cum putem configura și utiliza plăci de rețea în sistemul Linux.
Finnish-HOWTO.gz	Cum putem configura sistemul Linux pentru a utiliza limba finlandeză (cu excepția primului paragraf, întreg documentul este chiar în limba finlandeză)

Numele fișierului HOWTO	Conținutul
Firewall-HOWTO.gz	Cum putem configura un firewall Internet pe un sistem Linux.
French-HOWTO.gz	Cum putem configura sistemul Linux pentru a utiliza limba franceză (acest document este în limba franceză).
Ftape-HOWTO.gz	Cum putem configura și utiliza unități de bandă flexibilă (QIC-40, QIC-80, QIC-3010, QIC-3020 și compatibile cu acestea, conectate la PC prin intermediul controller-ului pentru discheta flexibilă) în Linux.
GCC-HOWTO.gz	Cum putem configura și utiliza compilatorul de C GNU și bibliotecile pentru dezvoltarea de aplicații în Linux.
German-HOWTO.gz	Cum putem utiliza setul de caractere german în Linux (acest document este în limba germană).
Glibc2-HOWTO.gz	Cum putem instala și utiliza biblioteca GNU pentru limbajul C, versiunea 2 (libc 6) în sistemele Linux.
HAM-HOWTO.gz	Informații despre programele utilizate de radioamatori în Linux.
Hardware-HOWTO.gz	Lista dispozitivelor fizice despre care se știe că funcționează sub Linux și modul în care putem găsi driverele necesare pentru acestea.
Hebrew-HOWTO.gz	Cum putem utiliza setul ebraic de caractere în sistemul X Window și în ferestrele din mod alfanumeric.
IPX-HOWTO.gz	Cum putem obține, instala și configura diferitele programe care utilizează protocolul IPX sub Linux (protocolul IPX provine din Novell Netware).
ISP-Hookup-HOWTO.gz	Cum putem conecta sistemul Linux la un furnizor de servicii Internet (ISP) utilizând un modem și o linie comutată.
Installation-HOWTO.gz	Cum putem obține și instala sistemul Linux.
Intranet-Server-HOWTO.gz	Cum putem utiliza sistemul Linux într-o rețea Intranet care dispune de sisteme UNIX, Novell Netware, Windows NT și Windows 95.
Italian-HOWTO.gz	Cum putem configura sistemul Linux pentru a utiliza limba italiană (acest document este chiar în limba italiană).
Java-CGI-HOWTO.gz	Cum putem dezvolta și utiliza programe Java în Linux.
Kernel-HOWTO.gz	Cum putem actualiza și reconstrui nucleul sistemului Linux.
Keyboard-and-Console-HOWTO.gz	Cum putem beneficia de diferitele utilitare Linux pentru configurarea tastaturii și consolei (în mod alfanumeric).
MGR-HOWTO.gz	Cum putem utiliza sistemul de ferestre în mod grafic MGR (ManaGeR) în Linux.
MILO-HOWTO.gz	Cum putem configura și utiliza programul Miniloader (MILO) în Linux pe sistemele cu procesoare Alpha AXP (la fel cum LILO încarcă și pornește sistemul Linux pe PC-urile cu procesoare Intel, MILO încarcă sistemul pe sistemele Alpha).
Mail-HOWTO.gz	Cum putem configura și întreține sistemul de poștă electronică (e-mail) într-un sistem Linux.
NET-3-HOWTO.gz	Cum putem instala și configura suportul pentru rețea TCP/IP în Linux.

(continuare)

Tabelul 1-6 Continuare

Numele fișierului HOWTO	Conținutul
NFS-HOWTO.gz	Cum putem configura un server și un client NFS (Network File System) în Linux.
NIS-HOWTO.gz	Cum putem configura serviciul NIS (Network Information Service) în Linux.
News-HOWTO.gz	Cum putem configura și întreține grupurile de discuții USENET în Linux.
Optical-Disk-HOWTO.gz	Cum putem instala și configura unități de disc optic în Linux (prezintă în detaliu unitatea Panasonic LF1000 PD Phase change cu interfața SCSI-II).
PCI-HOWTO.gz	Informații referitoare la suportul sistemului Linux pentru arhitectura magistralei PCI (Peripheral Component Interconnect)
PCMCIA-HOWTO.gz	Cum putem instala și utiliza plăci PCMCIA (Personal Computer Memory Card International Association) în Linux.
PPP-HOWTO.gz	Cum putem configura și utiliza protocolul PPP (Point-to-Point) în Linux.
Pilot-HOWTO.gz	Cum putem utiliza programul Palm Pilot Personal Digital Assistant (PDA) în sistemul Linux.
Polish-HOWTO.gz	Cum putem configura sistemul Linux pentru a utiliza limba poloneză (acest document este în limba poloneză).
Printing-HOWTO.gz	Cum putem configura serviciile de imprimare în Linux.
Printing-Usage-HOWTO.gz	Cum putem utiliza sistemul de virtualizare a imprimării din Linux.
RPM-HOWTO.gz	Cum putem utiliza programul Red Hat Package Manager (RPM) din Linux.
Reading-List-HOWTO.gz	O listă a cărților utile pentru cei interesați de Linux.
SCSI-HOWTO.gz	Informații referitoare la suportul pentru dispozitive SCSI din Linux.
SCSI-Programming-HOWTO.gz	Informații privind programarea driverelor SCSI în Linux (utile pentru programatorii care doresc să adauge suportul pentru un nou tip de dispozitiv SCSI în Linux)
SMB-HOWTO.gz	Cum putem utiliza sub Linux protocolul Server Message Block (SMB), cunoscut și sub numele de NetBIOS sau LAN Manager.
SRM-HOWTO.gz	Cum putem porni un sistem Linux cu procesor Alpha utilizând firmware-ul SRM (System Reference Manual), utilizat de regulă pentru inițializarea sistemului DEC UNIX pe procesoarele Alpha.
Serial-HOWTO.gz	Cum putem configura dispozitivele de comunicație serială în Linux.
Serial-Programming-HOWTO.gz	Cum putem programa porturile seriale în Linux.
Shadow-Password-HOWTO.gz	Cum putem obține, instala și configura sistemul Shadow Suite de protecție a parolelor în Linux (sistemul oferă un grad mai mare de protecție față de parolele memorate direct în fișierul /etc/passwd).
Slovenian-HOWTO.gz	Cum putem configura sistemul Linux pentru utilizatorii sloveni (acest document este în limba slovenă).

Numele fișierului HOWTO	Conținutul
Sound-HOWTO.gz	Cum putem activa suportul pentru placa de sunet în Linux.
Sound-Playing-HOWTO.gz	Lista tipurilor de formate de fișiere audio și a aplicațiilor din Linux care pot fi utilizate pentru a le asculta.
Spanish-HOWTO.gz	Cum putem configura sistemul Linux pentru utilizatorii vorbitori de spaniolă (acest document este în limba spaniolă).
TeX-HOWTO.gz	Cum putem instala și utiliza programele TeX și LaTeX pentru formatarea documentelor sub Linux.
Thai-HOWTO.gz	Cum putem configura sistemul Linux pentru limba thailandeză.
Tips-HOWTO.gz	Indicații și trucuri utile și totodată distractive.
UMSDOS-HOWTO.gz	Cum putem instala și utiliza sistemul de fișiere UMSDOS, care vă permite să instalați sistemul Linux într-un director MS-DOS (documentul nu este inclus în distribuția Red Hat Linux).
UPS-HOWTO.gz	Cum putem utiliza o sursă neîntreruptibilă de tensiune (UPS) în Linux.
UUCP-HOWTO.gz	Cum putem configura și utiliza programul UNIX-to-UNIX Copy (UUCP) în Linux.
User-Group-HOWTO.gz	Cum putem crea și întreține un grup de utilizatori Linux.
VAR-HOWTO.gz	Lista firmelor care revând distribuțiile sistemului Linux.
VMS-to-Linux-HOWTO.gz	Cum putem trece de la sistemul VMS la Linux (sistemul VMS rulează pe sistemele VAX produse de firma Digital Equipment Corporation).
XFree86-HOWTO.gz	Cum putem instala și configura XFree86 (sistemul X Window versiunea 11, distribuția 6 – X11R6) în Linux.
XFree86-Video-Timings-HOWTO.gz	Cum putem crea o linie de comandă în fișierul de configurare al serverului X pentru o anumită placă video (vezi capitolul 10 pentru o descriere mai amănunțită a liniei de comandă și a formatului fișierului de configurare pentru serverul X).

Oprirea sistemului Linux

De fiecare dată când sunteți pe punctul de a opri sistemul Linux, trebuie să acționați în mod ordonat. Chiar dacă sunteți singurul utilizator al sistemului Linux, alte câteva programe rulează de obicei în fundal. De asemenea, sistemele de operare (printre care și Linux) încearcă să realizeze o optimizare a modului în care scriu datele pe disc. Din cauză că accesul la disc este relativ lent (în comparație cu timpul necesar pentru accesul la locațiile de memorie), datele sunt de regulă păstrate în memorie, fiind apoi scrise pe disc în blocuri mari. Dacă pur și simplu opriți alimentarea calculatorului, riscați ca unele fișiere să nu fie actualizate corect.

Pentru a opri sistemul Linux, utilizați comanda shutdown. Este necesar să intrați în sistem ca root pentru a putea executa această comandă. Dacă ați intrat deja utilizând un alt nume, puteți deveni root tastând următoarea comandă (ceea ce introduceți dumneavoastră este prezentat îngroșat, comentariile autorului fiind scrise înclinat):

```
su
Password: (introduceți parola de root și apăsați Enter)
```

După ce ați devenit superutilizator, tastați următoarea comandă pentru a opri sistemul:

```
/sbin/shutdown -h now
```

Ca rezultat, va apărea următorul mesaj:

```
The system is going down for system halt NOW !!
```

După câteva momente, vor apărea multe alte mesaje care anunță oprirea proceselor aflate în rulare. În fine, va apărea mesajul:

```
System halted
```

Acum, puteți opri alimentarea în siguranță.

Dacă doriți doar să reporniți sistemul și nu să îl opriți, utilizați comanda shutdown cu opțiunea -r, după cum urmează:

```
/sbin/shutdown -r now
```



Sugestie

Un alt mod de repornire rapidă a sistemului este de a utiliza combinația de taste Ctrl+Alt+Delete – aceeași care se folosește pentru a reporni un PC și sub DOS. Oricine poate utiliza secvența Ctrl+Alt+Delete pentru a reporni un sistem Linux cu condiția să fie conectat la consola acestuia (adică la tastatura PC-ului, nu la un terminal conectat la un port serial).

Rezumat

Linux reprezintă un sistem de operare din familia sistemelor UNIX pentru procesoare de tipul Intel 80x86 sau compatibile cu acestea. CD-ROM-ul atașat lucrării include ultima versiune a binecunoscutei distribuții Red Hat Linux. Acest capitol vă prezintă pas cu pas procesul de instalare a sistemului Linux pe PC-ul dumneavoastră. Următorul capitol vă va învăța cum să mențineți nucleul sistemului de operare la zi și cum să instalați noi pachete de programe utilizând Red Hat Package Manager (RPM).

Acest capitol prezintă în detaliu următoarele probleme:

- ▶ Informațiile pe care trebuie să le dețineți despre PC-ul dumneavoastră înainte de a instala sistemul Linux
- ▶ Procesul de instalare a sistemului Red Hat Linux (inclusiv a sistemului X Window) de pe CD-ROM-ul atașat
- ▶ Pașii care trebuie executați sub MS-DOS sau Microsoft Windows înainte de a trece la instalarea sistemului Linux.
- ▶ Cum putem porni sistemul Linux inițial, partiționa hard-discul, iar apoi încărca diferitele componente de pe CD-ROM-ul atașat utilizând programul de instalare a sistemului Red Hat.
- ▶ Cum putem porni și opri un sistem Linux.

Capitolul 2

Actualizarea sistemului Linux

În acest capitol:

- ▶ Ce reprezintă nucleul sistemului Linux
- ▶ Utilizarea programului Red Hat Package Manager (RPM)
- ▶ Actualizarea nucleului prin aplicarea unui patch
- ▶ Regenerarea nucleului sistemului Linux
- ▶ Actualizarea nucleului la o nouă versiune utilizând RPM

Unul dintre motivele pentru care sistemul Linux trezește un interes sporit este acela că se îmbunătățește permanent prin contribuțiile aduse de un număr mare de programatori. Unii dintre acești programatori, spre exemplu, pot scrie drivere care adaugă suport pentru noi dispozitive fizice, cum ar fi o nouă unitate CD-ROM sau o nouă placă de rețea. Alți programatori pot contribui la extinderea numărului de plăci video care permit rularea sistemului X Window. Comunitatea care participă la dezvoltarea sistemului Linux pune aceste inovații la dispoziția tuturor sub forma unor pachete, cunoscute sub numele de patch-uri, sau prin intermediul noilor versiuni ale sistemului Linux.

Deși nu este necesar să actualizați sau să modificați sistemul Linux – mai precis nucleul acestuia – de fiecare dată când se lansează un nou patch sau o nouă versiune, actualizarea este totuși necesară din când în când, deoarece noile versiuni corectează unele probleme sau oferă performanțe mai bune în exploatarea dispozitivelor fizice din configurația dumneavoastră.

Este recomandat să reconstruiți nucleul sistemului chiar dacă nu apar corectări ale unor probleme sau îmbunătățiri majore. Nucleul sistemului Linux de pe CD-ROM-ul atașat lucrării este un nucleu generic, care include module ce permit utilizarea tuturor tipurilor de dispozitive fizice. Se recomandă să construiți un nucleu care să includă numai driverele corespunzătoare dispozitivelor fizice care se găsesc în configurația sistemului dumneavoastră. În particular, dacă aveți un disc SCSI, este necesar să creați un nucleu care să suporte adaptorul SCSI de care dispuneți. În funcție de necesități, puteți modifica anumite opțiuni din configurarea nucleului, cum ar fi activarea expedierii pachetelor IP, astfel încât să puteți utiliza sistemul Linux ca gateway Internet pentru rețeaua dumneavoastră locală (LAN).

Majoritatea corectărilor și îmbunătățirilor sunt oferite chiar de firma Red Hat – acestea sunt oferite sub formă de pachete Red Hat, fiind distribuite prin intermediul serverului FTP al firmei (<ftp://ftp.redhat.com/>). Pentru a instala aceste pachete, este necesar să utilizați programul Red Hat Package Manager.

Acest capitol începe cu un scurt ghid de utilizare a programului rpm – Red Hat Package Manager. În continuare, vom prezenta cum se poate aplica un patch asupra nucleului și cum se poate efectua reconstruirea acestuia.

Utilizarea programului Red Hat Package Manager

O inovație semnificativă adusă de distribuția Red Hat este *Red Hat Package Manager (RPM)* – un sistem care permite împachetarea tuturor fișierelor asociate unui produs software într-un singur fișier. Sistemul Red Hat Linux este distribuit sub forma unei colecții mari de fișiere RPM. Atunci când instalați sistemul Red Hat Linux de pe CD-ROM-ul atașat, programul de instalare utilizează comanda `rpm` pentru a desface pachetele, copiind fișierele în locațiile corespunzătoare de pe disc.

Chiar dacă nu aveți neapărat nevoie să cunoașteți structura internă a unui fișier RPM, este bine să știți cum puteți utiliza comanda `rpm` pentru lucrul cu astfel de fișiere. Mai precis, asupra fișierelor RPM puteți efectua următoarele operații:

- Determinarea numărului versiunii și a altor informații referitoare la fișierele RPM instalate în sistemul dumneavoastră.
- Instalarea unui nou pachet software dintr-un fișier RPM. De exemplu, puteți instala un pachet pe care l-ați omis la instalarea inițială. Ca un caz particular, este necesar să instalați fișierele sursă pentru nucleul sistemului Linux înainte de a trece la reconstruirea acestuia.
- Ștergerea (dezinstalarea) programelor care nu mai sunt necesare și pe care le-ați instalat anterior dintr-un fișier RPM. Dacă realizați că nu vă trebuie un pachet, este indicat să-l dezinstalați, pentru a refolosi spațiul pe disc ocupat de acesta.
- Actualizarea unei versiuni mai vechi a unui fișier RPM cu una mai nouă. Puteți efectua o actualizare după ce ați încărcat o versiune mai nouă a pachetului de pe serverul FTP al firmei Red Hat. În majoritatea cazurilor, este necesar să actualizați un fișier RPM pentru a beneficia de îmbunătățirile prezente în noua versiune.
- Verificarea integrității unui fișier RPM. Puteți verifica un anumit pachet cu scopul de a detecta dacă toate fișierele necesare se află la locurile potrivite.

Puteți efectua toate aceste activități utilizând comanda `rpm` și modificând doar diferitele opțiuni. În următoarele câteva secțiuni, vă vom prezenta pe scurt comanda `rpm`.



Sugestie

Dacă se întâmplă să uitați opțiunile comenzii `rpm`, tasteați următoarea comandă pentru a vedea o listă a acestora:

```
rpm --help|more
```

Veți fi chiar surprins de numărul mare al opțiunilor de care dispune comanda `rpm`!

Semantica numelor fișierelor RPM

Un fișier RPM conține un număr mare de fișiere, chiar dacă în sistemul Linux apare ca reprezentând unul singur. Prin convenție, numele fișierelor RPM au o structură specifică. Pentru a vizualiza numele fișierelor RPM de pe CD-ROM-ul atașat, procedați după cum urmează:

1. Introduceți CD-ROM-ul în unitate și montați unitatea utilizând comanda următoare:

```
mount /dev/hdc /cdrom
```

```
mount: block device /dev/hdc is write protected, mounting read-only
```

Asigurați-vă că în loc de `/dev/hdc` este scris numele unității de CD-ROM.

2. Deplasați-vă în directorul care conține fișierele RPM și vizualizați lista acestora:

```
cd /cdrom/RedHat/RPMS
ls *.rpm | more
```

```
AnotherLevel-0.5-2.noarch.rpm
BRU2000-15.0P-1.i386.rpm
BRU2000-X11-15.0P-1.i386.rpm
ElectricFence-2.0.5-5.i386.rpm
ImageMagick-3.9.1-1.i386.rpm
ImageMagick-devel-3.9.1-1.i386.rpm
MAKEDEV-2.3.1-1.noarch.rpm
SysVinit-2.71-3.i386.rpm
X11R6-contrib-3.3.1-1.i386.rpm
XFree86-100dpi-fonts-3.3.1-14.i386.rpm
XFree86-3.3.1-14.i386.rpm
XFree86-75dpi-fonts-3.3.1-14.i386.rpm
(am omis restul listei)
```

După cum v-ați dat probabil seama din listing, toate numele de fișiere RPM se termină cu extensia `.rpm`. Pentru a înțelege celelalte părți ale numelui unui fișier, să considerăm ca exemplu următorul fișier RPM:

```
XFree86-3.3.1-14.i386.rpm
```

Acest nume este compus din următoarele părți, separate între ele prin liniuțe (-):

- **Numele pachetului:** `XFree86`
- **Numărul versiunii:** `3.3.1`
- **Numărul de lansare:** `14` (acest număr este atribuit de către firma Red Hat)
- **Arhitectura:** `i386` (aceasta înseamnă că acest pachet este destinat procesoarelor compatibile cu Intel 80386)

De regulă, numele pachetului este suficient pentru ca să puteți intui ceea ce se află în conținutul fișierului RPM. Numărul versiunii coincide cu numărul curent al versiunii pachetului software (chiar și atunci când este distribuit într-un alt format, spre exemplu ca fișier `tar`). Firma Red Hat atribuie numerele de lansare pentru a ține mai ușor o evidență a modificărilor efectuate. Pentru RPM-uri pe care doriți să le instalați pe un PC cu un procesor compatibil cu Intel x86, arhitectura trebuie să fie `i386` sau `noarch`.

Informații despre fișierele RPM

Atunci când este utilizată pentru a instala pachete, comanda RPM construiește o bază de date a fișierelor RPM instalate. Puteți utiliza comanda `rpm -q` pentru a interoga această bază de date cu scopul de a afla informații despre pachetele instalate în sistemul dumneavoastră.

De exemplu, pentru a detecta numărul versiunii nucleului sistemului Linux instalat pe sistemul dumneavoastră, tasteați următoarea comandă `rpm -q`:

```
rpm -q kernel
Kernel-2.0.31-7
```

Comanda afișează numărul fișierului RPM pentru nucleu (ne referim aici la versiunea executabilă a nucleului și nu la fișierele sursă). Numele afișat este aproape la fel cu numele fișierului RPM, cu excepția părții finale `-i386.rpm` – care lipsește. În acest caz, partea care reprezintă versiunea fișierului RPM vă informează că nucleul are versiunea 2.0.31.

Dacă doriți să recompilați nucleul, este bine să cunoașteți ce versiune a compilatorului GNU pentru limbajul C (`gcc`) este instalată pe sistemul dumneavoastră. Pentru a vizualiza numărul versiunii programului `gcc`, tasteați:

```
rpm -q gcc
Gcc-2.7.2.3-8
```

De aici, puteți deduce că versiunea compilatorului gcc este 2.7.2.3.

Puteți vizualiza lista tuturor fișierelor RPM cu comanda:

```
rpm -qa
```

Ca rezultat, pe ecran va defila o listă lungă de fișiere RPM. Pentru a vizualiza lista ecran cu ecran, tastați comanda:

```
rpm -qa | more
```

Dacă doriți să căutați un anumit pachet, furnizați rezultatul comenzii rpm -qa ca intrare pentru comanda grep. De exemplu, pentru a vizualiza toate pachetele care conțin cuvântul kernel în numele lor, tastați:

```
rpm -qa | grep kernel
```

```
kernel-2.0.31-7
kernel-headers-2.0.31-7
kernel-modules-2.0.31-7
kernelcfg-0.4-7
Kernel-source-2.0.31-7
```

Puteți afla chiar mai mult decât numărul versiunii unui pachet utilizând comanda rpm -q. Adăugând opțiuni reprezentate printr-o singură literă, puteți afla și alte informații utile referitoare la un anumit pachet. De exemplu, puteți încerca următoarea comandă pentru a vizualiza fișierele conținute în pachetul care reprezintă nucleul sistemului Linux:

```
rpm -ql kernel
/boot/System.map-2.0.31
/boot/module-info-2.0.31
/boot/vmlinuz-2.0.31
/sbin/installkernel
```

În continuare, sunt listate câteva forme ale comenzilor de tip rpm -q, care permit aflarea informațiilor despre un anumit pachet (pentru a utiliza efectiv oricare dintre aceste comenzi rpm -q, tastați comanda urmată de numele pachetului):

rpm -qc	Listează toate fișierele de configurare dintr-un pachet.
rpm -qd	Listează toate fișierele de documentare dintr-un pachet. Acestea coincid de regulă cu paginile manualului online (cunoscute sub numele de pagini <i>man</i>).
rpm -qi	Afișează informații detaliate despre un pachet, printre care numărul versiunii, dimensiunea, data instalării precum și o scurtă descriere.
rpm -ql	Listează toate fișierele dintr-un pachet. Pentru unele pachete, lista afișată este destul de lungă.
rpm -qs	Listează starea tuturor fișierelor dintr-un pachet.

Instalarea unui fișier RPM

Pentru a instala un fișier RPM, trebuie să utilizați comanda rpm -i. Ca parametru al acestei comenzi se va furniza numele fișierului RPM. Un exemplu standard îl constituie instalarea unui fișier RPM de pe CD-ROM. De regulă, trebuie să montați mai întâi CD-ROM-ul, deplasându-vă apoi în directorul în care se găsesc fișierele RPM. Utilizați apoi comanda rpm -i pentru a instala fișierul RPM.

De exemplu, pentru a instala fișierul RPM care conține sursele nucleului sistemului de operare Linux de pe CD-ROM, se tastează secvența de comenzi:

```
mount /dev/hdc /cdrom
mount: block device /dev/hdc is write-protected, mounting read-only
cd /cdrom/RedHat/RPMS
rpm -i kernel-source*
```



Sugestie

Observați că nu este cazul să tastați numele întreg al fișierului RPM – puteți utiliza doar câteva caractere de la începutul numelui, urmate de un asterisc. Asigurați-vă însă că ați introdus o parte suficientă din numele fișierului pentru a-l identifica în mod unic.

Dacă veți încerca să instalați un fișier RPM deja instalat, comanda rpm va afișa un mesaj de eroare. De exemplu, iată ce se va întâmpla dacă vom încerca să reinstalăm editorul Emacs într-un sistem în care acesta există deja:

```
rpm -i emacs-20*
package emacs-20.2-4 is already installed
error: emacs-20.2-4.i386.rpm cannot be installed
```

Pentru a forța comanda rpm să instaleze un anumit pachet în ciuda erorilor, adăugați opțiunea --force la comanda rpm -i, ca în exemplul:

```
rpm -i --force emacs-20*
```

Eliminarea unui fișier RPM

Atunci când realizați că nu aveți nevoie de un anumit program, puteți proceda la eliminarea (dezinstalarea) pachetului asociat. De exemplu, dacă ați instalat pachetul care vă permite dezvoltarea de aplicații în sistemul X Window, dar nu vă interesează să scrieți aplicații pentru sistemul X, puteți elimina cu ușurință acest pachet cu ajutorul comenzii rpm -e.

Trebuie să cunoașteți exact numele unui pachet înainte de a trece la eliminarea acestuia. O modalitate eficientă de a determina numele unui pachet este utilizând comanda rpm -qa, în conjuncție cu grep pentru a căuta numele fișierului RPM corespunzător. De exemplu, pentru a determina fișierul RPM care permite dezvoltarea de aplicații sub sistemul X Window, putem utiliza secvența:

```
rpm -qa | grep XFree
XFree86-3.3.1-14
XFree86-75dpi-fonts-3.3.1-14
XFree86-devel-3.3.1-14
XFree86-libs-3.3.1-14
```

În acest caz, numele pachetului căutat este XFree86-devel. Pentru a elimina acest pachet, tastați comanda:

```
rpm -e XFree86-devel
```

După cum observați, nu este nevoie să introduceți numele complet al fișierului RPM; este suficient să cunoașteți numele pachetului – adică prima parte a numelui fișierului, până la prima liniuță aflată înaintea numărului versiunii.

Comanda rpm -e nu poate șterge un pachet de care alte pachete încă mai au nevoie. De exemplu, dacă vom încerca să eliminăm pachetul cu fișiere antet pentru nucleul sistemului, va apărea următorul mesaj de eroare:

```
rpm -e kernel-headers
removing these packages would break dependencies:
kernel-headers is needed by glibc-devel-2.0.5c-10
kernel-headers = 2.0.31 is needed by kernel-source-2.0.31-7
```


Actualizarea unui fișier RPM

Utilizați comanda `rpm -U` pentru a realiza actualizarea unui fișier RPM. Trebuie să introduceți ca parametru al comenzii numele fișierului RPM care conține noua versiune a programului. De exemplu, dacă avem instalată în sistem versiunea 1.2.4 a serverului Web Apache, dar dorim să actualizăm la versiunea 1.2.5, vom obține mai întâi fișierul RPM `apache-1.2.5-1.i386.rpm` de pe serverul FTP al firmei Red Hat (`ftp://ftp.redhat.com/pub/redhat/updates/5.0/i386`), utilizând ulterior comanda:

```
rpm -U apache-1.2.5-1.i386.rpm
```

Comanda `rpm` efectuează actualizarea, eliminând mai întâi vechea versiune a pachetului `apache` și instalând apoi noul fișier RPM.



Sugestie

Ori de câte ori este posibil, este recomandat să efectuați mai degrabă o actualizare decât să ștergeți vechiul pachet, după care să instalați noua versiune. Actualizarea salvează automat vechile fișiere de configurare, ceea ce vă scutește de neplăcerea de a configura programele după o nouă instalare.



Atenție!

Atunci când actualizați chiar nucleul sistemului de operare sau pachetele care conțin module, utilizați pentru instalare comanda `rpm -i` (în loc de `rpm -U`). În acest mod, nu veți distruge nucleul curent. Citiți secțiunea „Actualizarea nucleului utilizând un fișier RPM” pentru a învăța cum puteți instala un nou nucleu dintr-un fișier RPM.

Verificarea unui fișier RPM

Dacă suspectați că un anumit pachet nu a fost instalat corect (deși aceasta nu se va întâmpla prea des), utilizați comanda `rpm -V` pentru a verifica acel pachet. De exemplu, pentru a verifica pachetul care conține nucleul sistemului, tastați comanda:

```
rpm -V kernel
```

Ca urmare a acestei comenzi, programul `rpm` va compara mărimea și celelalte atribute ale fiecărui fișier din pachet cu cele ale fișierelor inițiale. Dacă verificarea decurge corect, comanda `rpm -V` nu va afișa nimic. Dacă există însă discrepanțe, va apărea un raport al acestora. De exemplu, să presupunem că am modificat fișierele de configurare pentru serverul de Web Apache. Iată ce se observă atunci când verificăm pachetul `apache`:

```
rpm -V apache
S.5....T c /etc/httpd/conf/access.conf
S.5....T c /etc/httpd/conf/httpd.conf
S.5....T c /etc/httpd/conf/srm.conf
```

În acest caz, rezultatul comenzii `rpm -V` ne arată că trei fișiere de configurare au fost modificate. Fiecare linie afișată este alcătuită din trei părți:

■ Linia începe cu opt caractere; fiecare dintre acestea indică tipul discrepanței detectate. De exemplu, `S` înseamnă că dimensiunea diferă, în timp ce `T` înseamnă că s-a modificat data ultimului acces la fișier. Tabelul 2-1 prezintă semnificația tuturor acestor caractere. Prezența unui punct indică faptul că atributul respectiv coincide cu cel inițial.

■ Pentru fișiere de configurare, apare în continuare un caracter `c`; în cazul altor fișiere, acest câmp este lăsat liber. În acest mod, puteți recunoaște cu ușurință fișierele de configurare. De regulă, nu este nevoie să vă îngrijorați de modificarea unui fișier de configurare, deoarece modificările au fost efectuate probabil chiar de către dumneavoastră.

■ Linia se încheie cu calea completă a fișierului. Citind această parte, puteți determina cu exactitate unde sunt localizate anumite fișiere.

Tabelul 2-1 Caracterele utilizate în rapoartele de verificare efectuate de RPM

Caracterul	Semnificația
S	Dimensiunea fișierului s-a modificat.
M	Permiisiunile de acces și tipul fișierului sunt diferite.
5	Suma de control calculată de algoritmul MD5 este diferită.
D	Tipul dispozitivului este diferit.
L	Legătura simbolică este diferită.
U	Proprietarul fișierului s-a modificat.
G	Grupul fișierului s-a modificat.
T	Data de acces la fișier s-a modificat.

Actualizarea nucleului prin aplicarea unor patch-uri

Ca orice produs de anvergură, sistemul Linux are un număr de versiune, care se modifică pe măsură ce sunt adăugate noi facilități. De regulă, numărul de versiune este compus din două părți: *numărul major al versiunii* și *numărul minor al versiunii*. Sistemul MS-DOS 6.2 are de exemplu un număr major al versiunii egal cu 6 și un număr minor egal cu 2. Numărul minor al versiunii se modifică pe măsură ce se efectuează îmbunătățiri incrementale ale programului. Numărul major este însă modificat numai la apariția unor modificări semnificative.

În plus față de numerele obișnuite ale unei versiuni, sistemul Linux este caracterizat și printr-un al treilea număr, denumit nivelul patch-ului curent. Patch-urile reprezintă de regulă corectări – modificări efectuate pentru a elimina anumite erori ale codului.

La pornirea sistemului Linux, mesajul introductiv vă prezintă atât numărul versiunii sistemului Linux cât și nivelul patch-ului. Este foarte posibil să apară o linie de forma:

```
kernel 2.0.31 on an i586
```

Ce reprezintă un patch al nucleului ?

Termenul de *nucleu* se referă la partea centrală a sistemului de operare Linux – acesta este programul care transformă un PC obișnuit într-un calculator rulând Linux. Termenul de *patch* se referă la corectări. Alăturând cei doi termeni, un *patch al nucleului* se referă la corecțiile care sunt aduse însuși sistemului de operare Linux.

Sistemul Linux este livrat împreună cu codul sursă; prin urmare, patch-urile reprezintă modificări ale

codului sursă. Atunci când aplicăm un patch al nucleului, modificăm de fapt codul sursă al sistemului de operare Linux. După efectuarea modificărilor, trebuie să reconstruim sistemul Linux, pentru ca acestea să devină vizibile. Reconstrucția nucleului înseamnă compilarea codului sursă al nucleului.

Acest mesaj ne spune că versiunea sistemului Linux este 2.0, iar nivelul patch-ului curent este 31. Versiunea 2.0.31 a nucleului a fost inițial inclusă pe CD-ROM-ul cu distribuția Red Hat 5.0.

Pentru copiile ulterioare ale sistemului Red Hat 5.0, firma Red Hat a actualizat nucleul la versiunea 2.0.32.

Puteți de asemenea să utilizați comanda `uname` pentru a determina numărul curent al versiunii nucleului Linux. Pentru aceasta, tastați:

```
uname -r
2.0.31
```

Rezultatul reprezintă numărul versiunii nucleului Linux. Comanda `uname` funcționează în orice alt sistem UNIX.

Deși numerele versiunilor se modifică relativ încet, o versiune cum este 2.0 conține relativ multe patch-uri. Puteți fi la curent cu apariția acestora și cu alte dezvoltări ale sistemului Linux citind periodic articolele grupurilor de discuții din USENET dedicate sistemului Linux (cum ar fi `comp.os.linux.announce` sau `comp.os.linux.setup`).



Referință
încrucișată

Este necesar să aveți acces la Internet pentru a putea citi grupurile de discuții din USENET. Consultați capitolele 18 și 19 pentru informații mai detaliate referitoare la conectarea sistemului dumneavoastră Linux la Internet.

Instalarea surselor nucleului Linux

Pentru a putea aplica patch-uri asupra nucleului, trebuie mai întâi să instalați fișierele care reprezintă sursele nucleului în sistemul dumneavoastră. Programul de instalare a sistemului Red Hat Linux nu efectuează instalarea acestor fișiere. Acestea sunt însă disponibile sub forma unui pachet Red Hat pe CD-ROM-ul atașat.

Procedați după cum urmează pentru a instala pachetul reprezentând sursele nucleului Linux în sistemul dumneavoastră:

1. Intrați în sistem ca `root` și introduceți CD-ROM-ul cu sistemul Red Hat Linux în unitatea CD.
2. Utilizați comanda `mount` pentru a monta unitatea CD-ROM într-un director al sistemului de fișiere. Formatul exact al acestei comenzi depinde de numele dispozitivului asociat unității CD-ROM și de directorul în care se efectuează montarea. De exemplu, pentru a monta un CD-ROM IDE (atașat pe cea de-a doua interfață IDE, care corespunde în Linux cu dispozitivul `/dev/hdc`) în directorul `/cdrom`, vom introduce comanda:

```
mount /dev/hdc /cdrom
mount: block device /dev/hdc is write protected, mounting read-only
```

Înlocuiți `/dev/hdc` cu numele dispozitivului care corespunde cu unitatea dumneavoastră de CD-ROM. Dacă directorul `/cdrom` nu există, introduceți comanda `mkdir /cdrom` pentru a-l crea.

3. Schimbați directorul curent de lucru cu directorul `RedHat/ RPMs` de pe CD-ROM și utilizați programul RPM pentru a instala fișierele care reprezintă sursele nucleului. Utilizați următoarele comenzi:

```
cd /cdrom/RedHat/RPMs
rpm -ivh kernel-source*
kernel-source
#####
```

Programul RPM indică starea desfășurării operației de instalare prin afișarea unei secvențe de semne diez (#). După ce RPM termină de instalat pachetul care reprezintă sursele nucleului, fișierele necesare se vor găsi în directorul `/usr/src/linux`.

Obținerea patch-urilor

Atunci când citiți că un anumit patch asigură o rezolvare a unora dintre problemele pe care le aveți cu sistemul Linux, de regulă aflați și numărul celui patch. Dacă veți cere ajutor într-o anumită problemă a sistemului Linux (cum ar fi „Rulez un nucleu versiunea 2.0.31, iar controllerul SCSI pe care l-am cumpărat nu funcționează”) prin trimiterea unor mesaje către un grup de discuții, este posibil să vi se răspundă că este nevoie să actualizați nucleul la versiunea 2.0.33, deoarece acel nou tip de controller SCSI este suportat începând cu versiunea 2.0.33.

Pentru actualizarea de la 2.0.31 la 2.0.33, aveți două soluții:

- Dacă sunt disponibile, obțineți sursele complete ale versiunii 2.0.33 a nucleului și instalați-le.
- Obțineți patch-urile 32 și 33 și aplicați-le la fișierele sursă din versiunea 2.0.31.



Sugestie

De regulă, distribuția completă a surselor nucleului reprezintă un fișier mare (de aproximativ 6 MB pentru versiunea 2.0.31) chiar în formă comprimată, în timp ce majoritatea patch-urilor au cel mult 100 KB (deși uneori pot ajunge până la 1 MB).

Merită prin urmare să încărcați din Internet și să aplicați doar patch-urile necesare. Încărcarea patch-urilor din Internet se face utilizând protocolul FTP (File Transfer Protocol).



Referință
încrucișată



Observație

Dacă nu aveți acces la Internet, capitolele 18 și 19 vă explică modul în care vă puteți conecta sistemul la Internet utilizând un modem (trebuie să vă faceți de asemenea un abonament la un furnizor de servicii Internet pentru a obține accesul).

Pentru a explica modul în care puteți obține patch-urile, vom presupune că sistemul dumneavoastră este conectat la Internet și că puteți utiliza comanda `ftp` pentru a avea acces la orice calculator din Internet.

Patch-urile nucleului Linux (ca de altfel și ultimele versiuni complete ale nucleului) se găsesc pe situl FTP `ftp.funet.fi` din Finlanda. Acest sit FTP organizează atât nuclele Linux, cât și patch-urile, în ordinea numărului versiunii. Toate fișierele pentru versiunea 2.0 sunt denumite după cum urmează (prezentăm doar câteva nume reprezentative):

```
linux-2.0.31.tar.gz (fișier comprimat, care conține nucleul complet, versiunea 2.0.31)
patch-2.0.32.gz (fișier comprimat, care conține patch-ul 32 pentru versiunea 2.0)
patch-2.0.33.gz (fișier comprimat, care conține patch-ul 33 pentru versiunea 2.0)
```

Fișierele Linux sunt distribuite în format comprimat, pentru a minimiza cantitatea de informație transmisă prin Internet. Sufixul `.gz` de la sfârșitul numelui unui fișier ne indică faptul că acel fișier a fost comprimat utilizând comanda `gzip`. După cum vom vedea ulterior, putem decompresa un astfel de fișier utilizând comanda `gunzip` (sau `gzip -d`).

Cea mai bună modalitate de a învăța cum să obțineți și să aplicați un patch este printr-un exemplu. În continuare, prezentăm ca exemplu modul în care putem obține fișierele patch relevante pentru trecerea de la versiunea Linux 2.0.31 la 2.0.33. Următoarea sesiune a programului `ftp` a utilizat o conexiune prin dial-up la Internet (viteza modemului fiind de 28800 de biți pe secundă, deoarece furnizorul de servicii nu utilizează încă modemuri de 56 Kbps):

```
ftp ftp.funet.fi
Connected to nic.funet.fi.220-Hello UNKNOWN at
dcc0xxx.slip.digex.net,220-220-Welcome to the FUNET archive, Please login as
'anonymous' with
220-your E-mail address as the password to access the archive.
220-See the README file for more information about this archive.
```

```

220-
220- All anonymous transfers are logged with your host name and whatever you
220- entered for the password. If you don't like this policy, disconnect now!
220-
220-
220-nic.funet.fi FTP server (Version 4.1452 problems@ftp.funet.fi ) ready.
220-There are 90 (max 250) archive users in your class at the moment.
220-Assuming 'login anonymous', other userids do vary.)
220-Local time is Sat Apr 4 02:25:26 1998 EET DST
220-
220 You can do "get README" even without logging in!
Name (ftp.funet.fi:naba): anonymous
331 Guest login ok, give your E-mail address for password.
Password:
230-Guest 'naba@access.digex.net' login ok.
230-
230- Finnish University and Research network FUNET
231- Archive FTP.FUNET.FI
230-
230-Most important file name suffixes are described at README.FILETYPES
230-News about this system can be looked at README.NEWS
230-
230-Welcome, you are 84th archive user in your class (max 250).
230-Your class is named: The known world outside NORDUnet region
230-There are 125 users in all classes (max 765)
230-Your data-transfer rate has no limitations.
230-
230-Local time is Sat Apr 4 02:26:33 1998 EET DST
230-
230-We have special access features, see file README
230- It was last updated Wed Nov 12 22:15:21 1997 - 142.1 days ago
230-
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/Linux/kernel/src/v2.0
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for.
-r--r--r-- 1 mirror mirror 26373 Aug 30 1996 Changes
-r--r--r-- 1 mirror mirror 0 Dec 16 22:55 LATEST-IS-2.03.33
-r--r--r-- 1 mirror mirror 16335 Jun 8 1996 Logo.gif
-r--r--r-- 1 mirror mirror 561 Jun 8 1996 Logo.txt
-r--r--r-- 1 mirror mirror 5859566 Jul 2 1996 linux-2.0.1.tar.gz
-r--r--r-- 1 mirror mirror 5940281 Jul 26 1996 linux-2.0.10.tar.gz
... (a portion of the listing is deleted)
-r--r--r-- 1 mirror mirror 6199356 Apr 7 1997 linux-2.0.30.tar.gz
-r--r--r-- 1 mirror mirror 6558735 Oct 17 21:25 linux-2.0.31.tar.gz
-r--r--r-- 1 mirror mirror 6567164 Nov 18 02:34 linux-2.0.32.tar.gz
-r--r--r-- 1 mirror mirror 6573183 Dec 16 22:55 linux-2.0.33.tar.gz
-r--r--r-- 1 mirror mirror 5926911 Jul 7 1996 linux-2.0.4.tar.gz
-r--r--r-- 1 mirror mirror 5927899 Jul 9 1996 linux-2.0.5.tar.gz
... (o parte din lista este intentionat omisa)
-r--r--r-- 1 mirror mirror 293692 Apr 7 1997 patch-2.0.30.gz
-r--r--r-- 1 mirror mirror 999854 Oct 17 21:25 patch-2.0.31.gz
-r--r--r-- 1 mirror mirror 41721 Nov 18 02:34 patch-2.0.32.gz
-r--r--r-- 1 mirror mirror 38497 Dec 16 22:55 patch-2.0.33.gz
-r--r--r-- 1 mirror mirror 47797 Jul 7 1996 patch-2.0.4.gz
-r--r--r-- 1 mirror mirror 14118 Jul 9 1996 patch-2.0.5.gz
-r--r--r-- 1 mirror mirror 6545 Jul 11 1996 patch-2.0.6.gz
-r--r--r-- 1 mirror mirror 9657 Jul 14 1996 patch-2.0.7.gz

```

```

-r--r--r-- 1 mirror mirror 40335 Jul 19 1996 patch-2.5.8.gz
-r--r--r-- 1 mirror mirror 27966 Jul 25 1996 patch-2.0.9.gz
drwxr-xr-x 35 mea mirror 8192 Feb 2 00:47 patch-html
226 Transfer complete.
ftp> binary
200 Type set to I.
ftp> get patch-2.0.32.gz
local: patch-2.0.32.gz remote: patch-2.0.32.gz
200 PORT command successful.
150 Opening BINARY mode data connection for
/pub/Linux/kernel/src/v2.0/patch-2.0.32.gz (41721 bytes)
226 Transfer complete.
41721 bytes received in 16 secs (2.6 Kbytes/sec)
ftp> get patch-2.0.33.gz
local: patch-2.0.33.gz remote: patch-2.0.33.gz
200 PORT command successful.
150 Opening BINARY mode data connection for
/pub/Linux/kernel/src/v2.0/patch-2.0.33.gz (38497 bytes)
226 Transfer complete.
38497 bytes received in 14.6 secs (2.6 Kbytes/sec)
ftp> bye
221-Goodbye, and thank you for using the FUNET archive.
221 You transferred 83 KBytes during this session.

```

De obicei, atunci când se transferă programe de pe un sit FTP, se utilizează numele de login *anonymous*. Tot de regulă, ca parolă se introduce adresa proprie de e-mail. Parcurgeți apoi următoarele etape:

1. Utilizați comanda `cd` pentru a schimba directorul curent de lucru în locația în care se găsesc fișierele versiunii curente a sistemului Linux.
2. Tastați comanda `ls -l` pentru a obține lista fișierelor din acel director. Căutați fișierele ale căror nume încep cu cuvântul *patch*; acestea sunt fișierele patch.
3. Tastați comanda `binary (bin)` pentru ca fișierele să fie transferate în mod binar.
4. Utilizați comanda `get` pentru a încărca fișierele patch necesare unul câte unul.
5. Tastați apoi una din comenzile `bye` sau `quit` pentru a părăsi programul `ftp`.

În continuare, treceți la despachetarea fișierelor patch și la aplicarea efectivă a patch-urilor.

Aplicarea patch-urilor

După ce ați transferat patch-urile, trebuie mai întâi să decompriți fișierele, după care le puteți aplica efectiv. Pentru a aplica patch-urile, procedați deci după cum urmează:

1. Copiați fișierele patch în directorul `/usr/src`.
2. Tastați următoarele comenzi pentru a ajunge în directorul `/usr/src` și a decompri fișierele patch:

```

cd /usr/src
gunzip patch*.gz

```
3. Utilizați comanda `patch` pentru a aplica patch-urile rând pe rând, după cum urmează:

```

patch -p0 < patch-2.0.32 2> err.32
patch -p0 < patch-2.0.33 2> err.33

```

Prima dintre aceste comenzi utilizează ca intrare standard fișierul `patch-2.0.32` și trimite toate mesajele afișate (inclusiv cele de eroare) în fișierul `err.32`.

După execuția celor două comenzi, verificați fișierele `err.32` și `err.33`, pentru a observa dacă vreunul dintre patch-uri nu a putut fi instalat. O modalitate bună de a efectua această verificare este utilizând instrumentul de căutare `grep` pentru a determina toate aparițiile șirului de caractere *fail*:

```
grep fail err.*
```

Dacă `grep` determină vreo apariție a șirului *fail*, înseamnă că patch-ul nu s-a instalat cu succes, cel mai probabil din cauză că fișierele sursă ale nucleului au fost deja modificate față de forma inițială.



Referință încrucișată

În acest moment, încercați doar această formă a comenzii `grep`. Capitolul 6 prezintă mai multe amănunte despre `grep` și alte comenzi ale sistemului Linux.

Dacă execuția comenzilor `patch` s-a desfășurat fără erori, codul sursă al sistemului de operare Linux va fi modificat, reflectând contribuțiile aduse de cei care dezvoltă sistemul ulterior lansării versiunii 2.0.31. După ultima din cele două comenzi, codul sursă al sistemului Linux a ajuns la nivelul 2.0.33. Este acum suficient să reconstruiți nucleul pentru ca sistemul Linux însuși să ajungă la versiunea 2.0.33.

Din nefericire, acești pași extrem de simpli nu se pot întotdeauna aplica, din cauză că firma Red Hat a modificat unele dintre fișierele componente. Spre exemplu, versiunea originală a sistemului Red Hat 5.0 a fost livrată cu un nucleu 2.0.31, dar unele dintre fișiere fuseseră deja modificate pentru versiunea 2.0.32. Atunci când veți încerca să aplicați patch-ul 2.0.32 la fișierele sursă 2.0.31, multe dintre fișiere vor apărea ca fiind deja instalate. Comanda `patch` vă va pune următoarea întrebare:

```
Reversed (or previously applied) patch detected! Assume -R? [Y]
```

Opțiunea `-R` este destinată tratării cazurilor în care fișierele din vechiul și noul patch au fost încorect amestecate. Puteți răspunde cu `No` pentru a sări un patch. Dacă aceasta se întâmplă, înseamnă că patch-urile nu se aplică în mod corect.



Sugestie

În cazul eșecului unei comenzi `patch`, puteți în continuare să efectuați actualizarea nucleului la o nouă versiune. Pentru aceasta, trebuie să obțineți noile versiuni de nucleu de pe situl FTP al firmei Red Hat. Puteți de asemenea să încărcați distribuția completă a surselor nucleului de la serverul `ftp.funet.fi` și apoi să reconstruiți nucleul, operație care va fi explicată în secțiunea următoare.

Regenerarea nucleului

Prin regenerarea unui nucleu înțelegem activitatea de creare a unui nou fișier binar pentru „inimă” sistemului de operare Linux. Acest fișier binar este încărcat la fiecare pornire a sistemului Linux. În general, necesitatea de a regenera nucleul poate apărea din următoarele motive:

- După ce ați aplicat patch-uri (așa cum s-a precizat în secțiunea precedentă), fișierele sursă ale sistemului de operare au fost actualizate, fiind necesară regenerarea nucleului și pornirea sistemului pentru a beneficia de noul nucleu.
- După instalarea inițială a sistemului Linux, este necesar să creați un nou nucleu care să includă suport numai pentru dispozitivele fizice care sunt instalate în sistemul dumneavoastră. Ca un caz particular, dacă dispuneți de un adaptor SCSI, va fi necesar să creați un nucleu care să conțină un driver SCSI. Nucleul de pe CD-ROM-ul atașat include driverele SCSI ca module externe care trebuie încărcate la pornire.

- După ce ați adăugat suportul pentru un nou dispozitiv (o placă de sunet, spre exemplu), trebuie să reconfigurați nucleul și să-l regenerați. În acest caz, veți selecta opțiunile corespunzătoare pentru a activa driverul pentru placa dumneavoastră de sunet.



Atenție!

Înainte de a regenera nucleul, pregătiți o dischetă de pornire în caz de urgență (vezi secțiunea „Pregătirea unei dischete de pornire în caz de urgență” din capitolul 1 pentru mai multe detalii). Dacă sistemul nu poate porni după ce nucleul a fost regenerat, puteți utiliza acea dischetă de pornire pentru a porni sistemul și a repeta procesul de construire a nucleului.

Pentru a regenera nucleul Linux, aveți nevoie de sursele nucleului. Sursele nucleului nu sunt instalate în mod prestabilit. Dacă nu ați instalat deja sursele, trebuie să le instalați acum, urmând instrucțiunile evidențiate anterior în secțiunea „Instalarea surselor nucleului”.

Generarea nucleului constă din următoarele faze principale:

- Configurarea nucleului
- Generarea nucleului
- Generarea modulelor
- Instalarea nucleului și a celorlalte fișiere necesare

Secțiunea următoare prezintă utilizarea modulelor în comparație cu includerea suportului pentru dispozitivele fizice direct în nucleu. Secțiunile următoare prezintă etapele construirii unui nucleu.

Utilizarea nucleului și a modulelor

Înainte de a începe configurarea nucleului, este bine să înțelegeți că dispuneți de două alternative în ceea ce privește instalarea driverele necesare pentru suportul diferitelor dispozitive fizice în Linux:

- *Suport inclus în nucleu:* Puteți include driverele pentru toate dispozitivele fizice din sistem direct în nucleu. După cum este ușor de înțeles, dimensiunea nucleului va crește proporțional cu numărul driverele pentru dispozitive pe care le încorporează. Un nucleu care include în interiorul său tot codul necesar pentru suportul dispozitivelor fizice se numește *nucleu monolitic*.
- *Utilizarea modulelor:* Puteți include driverele necesare sub forma unor module. Un *modul* este un bloc de cod pe care nucleul îl poate încărca după ce începe să ruleze. Modulele se utilizează de regulă pentru a adăuga suportul pentru noile dispozitive fizice fără a mai fi necesară o reconstruire a nucleului. Modulele nu sunt neapărat drivere pentru dispozitive; ele se pot de asemenea utiliza pentru a adăuga noi funcții la cele oferite de nucleu. Un nucleu care utilizează module se numește *nucleu modular*.

Nu este necesar să creați un nucleu complet monolitic sau complet modular. De obicei, suportul pentru o parte din dispozitive este legat direct în nucleu, în timp ce driverele pentru dispozitivele mai rar utilizate sunt construite sub formă de module.



Sugestie

Atunci când creați un nucleu pentru configurația dumneavoastră, puteți lega toate driverele necesare direct în nucleu. Dimensiunea unui astfel de nucleu monolitic poate fi păstrată în limite rezonabile, deoarece se includ numai driverele pentru dispozitivele fizice instalate efectiv în sistem.

Pentru o distribuție cum este cea a firmei Red Hat, se recomandă alegerea unui nucleu modular. Distribuția Red Hat pune la dispoziție un nucleu generic, împreună cu un număr mare de module, astfel încât să suporte toate tipurile de dispozitive fizice. Programul de instalare va

configura apoi sistemul Linux astfel încât să încarce numai modulele necesare pentru suportul dispozitivelor instalate efectiv în sistemul utilizatorului.

Atunci când configurați nucleul, trebuie să alegeți modul în care va fi inclus suportul pentru diferitele dispozitive fizice. De regulă, pentru fiecare dintre opțiunile de configurare, trebuie să răspundeți cu una din următoarele variante:

- **y** pentru a include suportul direct în nucleu
- **m** pentru a utiliza un modul
- **n** pentru a nu include suport pentru acel dispozitiv

În caz că un anumit dispozitiv nu are un driver modular, opțiunea **m** nu este prezentă.

Configurarea nucleului

Pentru a configura nucleul, intrați în sistem ca **root**. Deplasați-vă apoi în directorul `/usr/src/linux` utilizând comanda:

```
cd /usr/src/linux
```

Configurarea nucleului constă în a indica facilitățile generale și driverele pentru dispozitive pe care doriți să le includeți în sistemul Linux. În esență, veți construi o copie a sistemului Linux, în care veți amesteca acele facilități pe care le doriți.

Sistemul Red Hat Linux vă oferă trei moduri în care puteți configura nucleul:

- Tastați `make menuconfig` pentru a introduce parametri de configurare a nucleului cu ajutorul unei interfețe în mod text, similară cu cea utilizată de programul de instalare a sistemului Red Hat.
- Tastați `make xconfig` pentru a utiliza un program de configurare cu o interfață X Window. Pentru a utiliza acest program de configurare în mod grafic, trebuie să rulați serverul X.
- Tastați `make config` pentru a utiliza un program care vă cere să introduceți opțiunile de configurare una după cealaltă. Puteți utiliza acest program de configurare direct de la promptul de comandă al sistemului Linux.

Toate aceste programe conduc la obținerea aceluiași rezultat – fiecare memorează opțiunile introduse într-un fișier text numit `.config`, care se găsește în directorul `/usr/src/linux`. Din cauză că numele acestui fișier începe cu un punct, fișierul nu va fi vizibil atunci când veți utiliza comanda `ls` pentru a lista conținutul directorului. Tastați `ls -a` pentru a vedea acest fișier în directorul în care se găsește.



Observație

Rezultatul pasului de configurare este înregistrarea opțiunilor dumneavoastră în fișierul `.config`. Fișierul care reprezintă efectiv nucleul nu se va modifica până în momentul în care veți compila un nou nucleu utilizând comanda `make`. În

consecință, puteți parcurge etapa de configurare a nucleului ori de câte ori doriți.

Iată câteva linii dintr-un fișier `.config` (obținut după parcurgerea etapei de configurare):

```
more .config
#
# Automatically generated make config; don't edit
#
#
# Code maturity level options
#
# CONFIG_EXPERIMENTAL is not set
#
```

```
# Loadable module support
#
CONFIG_MODULES=y
# CONFIG_MODVERSIONS is not set
CONFIG_KERNEL=y

#
# General setup
#
# CONFIG_MATH_EMULATION is not set
CONFIG_NET=y
# CONFIG_MAX_16M is not set
CONFIG_PCI=y
CONFIG_SYSVIPC=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_ELF=y
CONFIG_KERNEL_ELF=y
# CONFIG_M386 is not set
# CONFIG_M486 is not set
CONFIG_M586=y
# CONFIG_M686 is not set
(omitem prezentarea restului fisierului)
```

În esență, fiecare opțiune de configurare are un nume și fiecărei opțiuni i se atribuie o valoare. Numele unei opțiuni începe cu `CONFIG_`, urmat de un cuvânt prin care se identifică opțiunea. Fiecare opțiune selectată are valoarea **y** (ceea ce înseamnă că suportul pentru facilitatea respectivă va fi încorporat direct în nucleu) sau **m** (în cazul în care facilitatea va fi implementată printr-un modul). Liniile care încep cu caracterul `#` reprezintă comentarii. Facilitățile care nu au fost selectate sunt reprezentate prin linii comentate.

În exemplul de mai sus, opțiunea `CONFIG_PCI` are valoarea **y**, ceea ce înseamnă că nucleul va include suport pentru dispozitive conectate la magistrala PCI (Peripheral Component Interconnect). Pe de altă parte, opțiunea `CONFIG_MATH_EMULATION` nu este activată – aceasta înseamnă că nucleul nu are nevoie de cod pentru emularea unui coprocesor matematic (procesoarele 386 și 486SX au nevoie de un coprocesor matematic separat, în timp ce procesoarele mai noi, cum sunt Intel Pentium și AMD K6 înglobează un coprocesor matematic).



Sugestie

Vom descrie în continuare operația de configurare cu ajutorul comenzii `make config`. Deși această metodă este relativ lentă, deoarece opțiunile sunt parcurse una câte una, este ideală pentru scopuri didactice. Pe măsură ce vom trece de la un grup de opțiuni la altul, vom prezenta note care vor explica semnificația opțiunilor din grup. Pentru a efectua configurarea nucleului, puteți utiliza orice instrument de configurare (`make xconfig`, `make menuconfig` sau `make config`).

Pentru a începe operația de configurare, tastați comanda:

```
make config
rm -f include/asm (cd include; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
```

Prima întrebare se referă la includerea codului experimental sau a driverelor pentru dispozitive aflate în stadiul de experiment:

```
** Code maturity level options * Prompt for development
and/or incomplete code drivers (CONFIG_EXPERIMENTAL)
[Y/n/?] n
```


În majoritatea cazurilor (cu excepția situației în care căutați neapărat aventura) puteți răspunde cu No la această întrebare.



Sugestie

Răspunsurile posibile la fiecare întrebare apar incluse în paranteze pătrate, răspunsul prestabilit fiind scris cu literă mare. Prin urmare, [Y/n/?] înseamnă că răspunsul prestabilit este Yes, celelalte două posibile fiind n și ?. Pentru a accepta răspunsul prestabilit, apăsați Enter.

Următorul grup de opțiuni se referă la suportul pentru modulele pe care nucleul le poate încărca (modulele sunt programe similare programelor rezidente – TSR din sistemul MS-DOS):

* Loadable module support

*

Enable loadable module support (CONFIG_MODULES) [Y/n/?] y

Set version information on all symbols for modules

(CONFIG_MODVERSIONS) [Y/n/?] n

Kernel daemon support (e.g. autoload of modules)

(CONFIG_KERNELD) [Y/n/?] y

Suportul pentru module este necesar în majoritatea cazurilor, deci se recomandă să răspundeți cu Yes la prima întrebare.

Următoarea întrebare se referă la includerea informației care reprezintă versiunea pentru fiecare modul. Atunci când informația referitoare la versiune este prezentă la un modul, se efectuează o verificare a compatibilității modulului cu versiunea curentă a nucleului. Din cauză că eliminarea unui modul care nu funcționează se face ușor, se recomandă să răspundeți cu No la această întrebare. Puteți însă să acceptați cu siguranță opțiunea implicită, apăsând Enter.

Cea de-a treia întrebare se referă la demonul (daemon) nucleului – un program care rulează continuu și încarcă modulele de care nucleul are nevoie. Trebuie să răspundeți cu Yes la această întrebare.

Urmează un set de opțiuni generale, începând cu întrebarea:

*

* General setup

Kernel math emulation (CONFIG_MATH_EMULATION) [Y/n/?] n

Dacă aveți un procesor mai vechi 386 sau 486SX, care nu are coprocesor matematic, puteți răspunde cu Yes la această întrebare, iar nucleul va efectua operațiile aritmetice în virgulă mobilă prin software.

Următoarea opțiune are ca subiect activarea suportului pentru rețea:

Networking support (CONFIG_NET) [Y/n/?] y

Este necesar să răspundeți cu Yes la această întrebare chiar dacă nu aveți o placă de rețea sau nu intenționați să conectați sistemul la Internet. Aceasta din cauză că unele din programele care rulează în sistem au nevoie de o conexiune la o rețea internă pentru a comunica. Aveți nevoie de suportul pentru rețeaua internă pentru ca totul să funcționeze corect.

Urmează o întrebare referitoare la limitarea memoriei sistemului la 16MB:

Limit memory to low 16MB (CONFIG_MAX_16M) [N/y/?] n

Motivul prezenței acestei întrebări este faptul că multe PC-uri mai vechi nu pot utiliza mai mult de 16 MB de memorie. Dacă aveți un astfel de sistem (sau dacă memoria sistemului dumneavoastră este mai mică de 16 MB), trebuie să răspundeți cu Yes. Pentru calculatoarele PC mai noi, se recomandă să răspundeți cu No.

Următoarea opțiune de configurare vă permite activarea suportului pentru magistrala PCI:

PCI bios support (CONFIG_PCI) [Y/n/?] y

Dacă PC-ul dumneavoastră are o magistrală PCI (majoritatea noilor calculatoare PC Pentium au o magistrală PCI), răspundeți cu Yes.

Următoarea întrebare se referă la suportul pentru intercomunicația între procese (IPC) utilizând mecanismele System V:

System V IPC (CONFIG_SYSVIPC) [Y/n/?] y

Ca și în cazul suportului pentru rețea, mecanismele System V (este cazul versiunii System V a sistemelor UNIX) IPC sunt utilizate de programe pentru a comunica între ele. Trebuie să activați această opțiune, răspunzând cu Yes.

Următoarele câteva întrebări au ca subiect formatul fișierelor binare și tipul procesorului:

Kernel support for a.out binaries (CONFIG_BINFMT_AOUT)

[Y/m/n/?] y

Kernel support for ELF binaries (CONFIG_BINFMT_ELF)

[Y/m/n/?] y

Compile kernel as ELF – if your GCC is ELF-GCC

(CONFIG_KERNEL_ELF) [Y/n/?] y

Processor type (386, 486, Pentium, PPro) [386] Pentium

defined CONFIG_M586

Formatul a.out reprezintă un format mai vechi al fișierelor executabile (denumite de asemenea *fișiere binare*). Trebuie să selectați Yes, deoarece veți rula și câteva programe mai vechi, care utilizează formatul a.out.

ELF (Executable and Linking Format) reprezintă noul format standard al programelor executabile. În mod sigur, nucleul trebuie să suporte binarele ELF.

Nucleul însuși trebuie compilat și memorat în format ELF; prin urmare, răspundeți cu Yes la această a treia întrebare.

Ultima întrebare din acest set vă cere să specificați tipul procesorului din sistem. Dacă răspundeți 386, nucleul compilat va putea rula pe orice procesor, inclusiv pe 486 sau Pentium. Dacă însă doriți să creați un nucleu special pentru sistemul dumneavoastră, introduceți exact tipul procesorului.

Următorul set de întrebări se referă la unitatea de dischetă și la cea de hard-disc. Prima întrebare se referă la suportul pentru unitatea de disc:

*

* Floppy, IDE, and other block devices

*

Normal floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?] y

Cum majoritatea PC-urilor au unitate de dischetă, trebuie să răspundeți cu Yes.

Trebuie să răspundeți cu Yes și la următoarea întrebare dacă dispuneți în sistem de una sau mai multe unități IDE (Integrated Drive Electronics):

Enhanced IDE/MFM/RLI disk/cdrom/tape/floppy support

(CONFIG_BLK_DEV_IDE) [Y/n/?] y

*

* Please see Documentation/ide.txt for help/info on IDE drives

*



Sugestie

Nota din mesajul anterior se referă la fișierul /usr/src/linux/Documentation/ide.txt.

În directorul /usr/src/linux/Documentation se găsesc multe fișiere utile cu rol de documentare, toate în format text.

Următoarea întrebare se referă la un driver IDE vechi care suportă numai hard-discurile (noile drivere IDE suportă și alte dispozitive, cum ar fi unitățile CD-ROM). Trebuie să răspundeți cu No, cu excepția situației în care dispuneți de un PC mai vechi, care permite conectarea la interfața IDE numai a unei unități de hard-disc:

Use old disk-only driver on primary interface
(CONFIG_BLK_DEV_HD_IDE) [N/y/?] **n**

Dacă aveți o unitate CD-ROM IDE, răspundeți cu Yes la următoarea întrebare (ATAPI este un alt nume pentru varianta îmbunătățită a interfeței IDE):

Include IDE/ATAPI CDRom support (CONFIG_BLK_DEV_IDECD)
[Y/n/?] **y**

Dacă aveți o unitate de bandă conectată la interfața IDE, trebuie să răspundeți cu Yes la întrebarea următoare. Dacă nu dispuneți de o unitate de bandă, răspundeți cu No:

Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE)
[Y/n/?] **n**

Dacă aveți o unitate de dischetă conectată la interfața IDE, cum este unitatea Zip Iomega, răspundeți cu Yes la întrebarea:

Include IDE/ATAPI FLOPPY support (new)
(CONFIG_BLK_DEV_IDEFLOPPY) [Y/n/?] **y**

Următoarea opțiune vă permite să utilizați un driver SCSI pentru accesul la un dispozitiv conectat la interfața IDE. În majoritatea cazurilor, este cazul să răspundeți cu No la această întrebare:

SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N/y/?] **n**

Suportul pentru interfețele IDE construite pe plăcile de tip PC Card poate fi activat selectând următoarea opțiune. Răspundeți cu No la această întrebare, cu excepția cazului în care dispuneți de o astfel de placă:

Support removable IDE interfaces (PCMCIA)
(CONFIG_BLK_DEV_IDE_PCMCIA) [Y/n/?] **n**

Următoarele două opțiuni corectează anumite probleme prezente la anumite cipseturi utilizate la interfețele IDE. Răspundeți cu Yes la ambele întrebări:

CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y/n/?] **y**
CMD640 enhanced support (CONFIG_BLK_DEV_CMD640_ENHANCED) [Y/n/?] **y**
RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ1000) [Y/n/?] **y**

Următoarea întrebare se referă la un sistem cu magistrală PCI, care dispune de o unitate de disc IDE care utilizează de asemenea cipsetul Intel Triton I/II. Dacă nu cunoașteți acest detaliu referitor la sistemul dumneavoastră, este mai sigur să răspundeți cu Yes la următoarea întrebare:

Intel 82371 PIIX (Triton I/II) DMA support
(CONFIG_BLK_DEV_TRITON) [Y/n/?] **y**

Răspundeți cu Yes la întrebarea următoare dacă doriți să activați suportul pentru diversele cipseturi IDE utilizate pe plăcile de bază care se găsesc în majoritatea PC-urilor. Acest suport este necesar pentru ca sistemul Linux să poată utiliza a treia și respectiv a patra unitate IDE. Dacă dispuneți de mai mult de două unități IDE, este necesar să activați această opțiune. În exemplu, am răspuns cu No, deoarece am presupus că majoritatea sistemelor dispun de numai două unități IDE – o unitate de hard-disc și o unitate CD-ROM:

Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?] **n**

Următoarea opțiune permite suportul pentru dispozitivul de loopback – acesta permite sistemului Linux să poată manipula un întreg sistem de fișiere prin intermediul unui singur fișier mare. Puteți răspunde cu No la această întrebare:

*
* Additional Block Devices
*
Loopback device support (CONFIG_BLK_DEV_LOOP) [M/n/y/?] **n**

Dacă doriți să combinați mai multe partiții ale hard-discului într-un singur dispozitiv logic, răspundeți cu Yes la următoarea întrebare. Dacă nu sunteți sigur, se recomandă să răspundeți cu No: Multiple devices driver support (CONFIG_BLK_DEV_MD) [Y/n/?] **n**

Dacă doriți să utilizați o parte a memoriei sistemului cu rolul de disc capabil de a memora un sistem de fișiere, răspundeți cu Yes la întrebarea care urmează. De regulă, un disc RAM este util numai la pornirea sistemului, atunci când hard-discul nu este încă disponibil. Puteți răspunde cu m, iar în acest caz suportul pentru discul RAM va fi implementat sub forma unui modul: RAM disk support (CONFIG_BLK_DEV_RAM) [Y/m/n/?] **m**

Răspundeți cu Yes la întrebarea următoare numai dacă PC-ul dumneavoastră utilizează un controller mai vechi pentru hard-disc, de tipul celor care erau prezente în structura calculatoarelor IBM XT la mijlocul deceniului trecut:

XT harddisk support (CONFIG_BLK_DEV_XD) [N/y/m/?] **n**

Următorul set de opțiuni se referă la configurarea rețelei. Răspunsurile depind de maniera în care doriți să utilizați sistemul Linux în cadrul unei rețele. Iată câteva indicații:

■ Răspundeți cu Yes la opțiunea CONFIG_FIREWALL dacă doriți să utilizați sistemul Linux ca firewall. Un firewall reprezintă un sistem intermediar care controlează fluxul de informație vehiculat între rețeaua locală (LAN) și Internet. Această opțiune trebuie activată și dacă doriți să utilizați deghizarea IP (IP masquerading), care permite mai multor sisteme dintr-o rețea LAN privată să aibă acces la Internet prin intermediul unui sistem Linux cu o adresă IP validă. În acest caz este necesar să activați și opțiunea CONFIG_IP_FORWARD. Deghizarea IP este explicată amănunțit în capitolul 16.

■ Dacă răspundeți cu Yes la opțiunea CONFIG_NET_ALIAS, veți putea atribui adrese IP multiple unei singure plăci de rețea.

■ Trebuie să activați întotdeauna opțiunea CONFIG_INET pentru a utiliza protocoalele de rețea TCP/IP.

■ Activând opțiunea IP forwarding (CONFIG_IP_FORWARD), veți putea utiliza sistemul Linux ca router – sistem aflat la granița a două rețele LAN și care permite comunicația între cele două rețele.

■ Trebuie de asemenea să activați opțiunea CONFIG_IP_MULTICAST dacă doriți să utilizați sistemul ca router.

■ Se recomandă să activați opțiunea CONFIG_SYN_COOKIES dacă doriți ca sistemul Linux să poată rezista la o categorie de atacuri cunoscute sub numele de SYN flooding, care pot determina supraîncărcarea rețelei.

■ Răspundeți cu Yes la CONFIG_IP_ACCT numai dacă doriți să utilizați sistemul Linux ca router.

■ Dacă utilizați sistemul ca router, activați de asemenea și opțiunea CONFIG_IP_ROUTER.

* Puteți răspunde cu Yes la opțiunea IP tunneling (CONFIG_NET_IPIP). Este necesar să modificați această opțiune numai în anumite cazuri, cum ar fi deplasarea unui calculator laptop dintr-o rețea în alta fără a-i modifica adresa IP.

Iată răspunsurile pentru care puteți opta la configurarea sistemului dumneavoastră (presupunând că nu veți utiliza sistemul ca router):

```
*
* Networking options
*
Network firewalls (CONFIG_FIREWALL) [Y/n/?] n
Network aliasing (CONFIG_NET_ALIAS) [Y/n/?] n
TCP/IP networking (CONFIG_INET) [Y/n/?] y
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?] n
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] n
IP: syn cookies (CONFIG_SYN_COOKIES) [Y/n/?] y
IP: accounting (CONFIG_IP_ACCT) [Y/n/?] n
IP: optimize as router not host (CONFIG_IP_ROUTER) [N/y/?] n
IP: tunneling (CONFIG_NET_IPIP) [M/n/y/?] n
```

Următorul set de opțiuni se referă de asemenea la rețea. După cum indică și comentariul care le precede, se recomandă să acceptați valorile implicite pentru aceste opțiuni (apăsăți Enter pentru a selecta valorile prestabilite):

```
*
* (it is safe to leave these untouched)
*
IP: PC/TCP compatibility mode (CONFIG_INET_PCTCP) [N/y/?] n
IP: Reverse ARP (CONFIG_INET_RARP) [M/n/y/?] n
IP: Disable Path MTU Discovery (normally enabled)
(CONFIG_NO_PATH_MTU_DISCOVERY) [N/y/?] n
IP: Drop source routed frames (CONFIG_IP_NOSR) [Y/n/?] n
IP: Allow large windows (not recommended if <16Mb of
memory) (CONFIG_SKB_LARGE) [Y/n/?] n
```

Cu ajutorul următoarelor patru opțiuni, puteți activa suportul pentru alte protocoale în afară de TCP/IP. Protocoalele suportate sunt IPX (protocolul utilizat de sistemul Novell Netware), Appletalk și AX.25 (utilizat în rețelele cu comunicația prin radio). Cea de-a patra opțiune permite efectuarea unei legături prin intermediul rețelei între nucleu și alte procese. Vom dezactiva toate aceste opțiuni, întrucât în majoritatea situațiilor nu avem nevoie de ele:

```
*
*
*
The IPX protocol (CONFIG_IPX) [M/n/y/?] n
Appletalk DDP (CONFIG_ATALK) [M/n/y/?] n
Amateur Radio AX.25 Level 2 (CONFIG_AX25) [N/y/?] n
Kernel/User network link driver (CONFIG_NETLINK) [N/y/?] n
```

Urmează apoi un întreg set de opțiuni legate de dispozitivele SCSI. Dacă sistemul dumneavoastră are un controller SCSI, trebuie să răspundeți cu Yes la opțiunea CONFIG SCSI. După aceasta, urmează să răspundeți întrebărilor referitoare la tipurile de dispozitive SCSI (disc, bandă, CD-ROM) conectate la adaptorul SCSI. În cele din urmă, trebuie să activați suportul pentru modelul specific de adaptor SCSI prezent în configurația sistemului.



Sugestie

Dacă sistemul dumneavoastră are un adaptor SCSI, selectați întotdeauna y la opțiunile SCSI necesare. Nu selectați m pentru a crea module SCSI. Instalarea unui nucleu care trebuie să încarce drivere SCSI modulare necesită câțiva pași în plus. Cea mai simplă metodă este de a încorpora în nucleu suportul pentru dispozitivele SCSI, răspunzând cu y la opțiuni cum sunt CONFIG SCSI, CONFIG_BLK_DEV_SD, cât și la opțiunea specifică adaptorului SCSI din configurația dumneavoastră (puteți consulta lista prezentată în continuare).

Lista următoare prezintă opțiunile necesare pentru configurarea suportului SCSI, adaptorul folosit fiind de tipul Adaptec AHA 1542:

```
*
* SCSI support
*
SCSI support (CONFIG SCSI) [N/y/m/?] y
*
* SCSI support type (disk, tape, CD-ROM)
*
SCSI disk support (CONFIG_BLK_DEV_SD) [N/y/m/?] (NEW) y
SCSI tape support (CONFIG_CHR_DEV_ST) [N/y/m/?] (NEW) n
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [N/y/m/?] (NEW) n
SCSI generic support (CONFIG_CHR_DEV_SG) [N/y/m/?] (NEW) y
*
* Some SCSI devices (e.g. CD jukebox) support multiple LUNs
*
Probe all LUNs on each SCSI device (CONFIG SCSI_MULTI_LUN) [N/y/?] (NEW) n
Verbose SCSI error reporting (kernel size >=12K) (CONFIG SCSI_CONSTANTS) [N/y/?] (NEW) n
*
* SCSI low-level drivers
*
7000FASST SCSI support (CONFIG SCSI_7000FASST) [N/y/m/?] (NEW)
Adaptec AHA152X/2825 support (CONFIG SCSI_AHA152X) [N/y/m/?] (NEW)
Adaptec AHA1542 support (CONFIG SCSI_AHA1542) [N/y/m/?] (NEW) y
Adaptec AHA1740 support (CONFIG SCSI_AHA1740) [N/y/m/?] (NEW)
Adaptec AIC7xxx support (CONFIG SCSI_AIC7XXX) [N/y/m/?] (NEW)
AdvanSys SCSI support (CONFIG SCSI_ADVANSYS) [N/y/m/?] (NEW)
Always IN2000 SCSI support (CONFIG SCSI_IN2000) [N/y/m/?] (NEW)
AM53/79C974 PCI SCSI support (CONFIG SCSI_AM53C974) [N/y/m/?] (NEW)
BusLogic SCSI support (CONFIG SCSI_BUSLOGIC) [N/y/m/?] (NEW)
DTC3180/3280 SCSI support (CONFIG SCSI_DTC3280) [N/y/m/?] (NEW)
EATA-DMA (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support
(CONFIG SCSI_EATA_DMA) [N/y/m/?] (NEW)
EATA-PIO (old DPT PM2001, PM2012A) support (CONFIG SCSI_EATA_PIO) [N/y/m/?] (NEW)
EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support
(CONFIG SCSI_EATA) [N/y/m/?] (NEW)
Future Domain 16xx SCSI support (CONFIG SCSI_FUTURE_DOMAIN) [N/y/m/?] (NEW)
Generic NCR5380/53c400 SCSI support (CONFIG SCSI_GENERIC_NCR5380) [N/y/m/?] (NEW)
NCR53c406a SCSI support (CONFIG SCSI_NCR53C406A) [N/y/m/?] (NEW)
NCR53c7,8XX SCSI support (CONFIG SCSI_NCR53C7xx) [N/y/m/?] (NEW)
NCR53C8XX SCSI support (CONFIG SCSI_NCR53C8XX) [N/y/m/?] (NEW)
IOMEGA Parallel Port ZIP drive SCSI support (CONFIG SCSI_PPA) [N/y/m/?] (NEW)
PAS16 SCSI support (CONFIG SCSI_PAS16) [N/y/m/?] (NEW)
Qlogic FAS SCSI support (CONFIG SCSI_QLOGIC_FAS) [N/y/m/?] (NEW)
Qlogic ISP SCSI support (CONFIG SCSI_QLOGIC_ISP) [N/y/m/?] (NEW)
Seagate ST-02 and Future Domain TMC-8xx SCSI support (CONFIG SCSI_SEAGATE)
[N/y/m/?] (NEW)
Tekram DC-390(T) SCSI support (CONFIG SCSI_DC390T) [N/y/m/?] (NEW)
Trantor T128/T128F/T228 SCSI support (CONFIG SCSI_T128) [N/y/m/?] (NEW)
UltraStor 14F/34F support (CONFIG SCSI_U14_34F) [N/y/m/?] (NEW)
UltraStor SCSI support (CONFIG SCSI_ULTRASTOR) [N/y/m/?] (NEW)
GDT SCSI Disk Array Controller support (CONFIG SCSI_GDTH) [N/y/m/?] (NEW)
```

Răspundeți cu Yes la următoarea întrebare dacă în sistem este instalată o placă de rețea:

```
*
* Network device support
*
```

```
Network device support (CONFIG_NETDEVICES) [Y/n/?] y
```

Următoarea opțiune, atunci când este activată, creează un dispozitiv fictiv, care este necesar pentru conexiunile inactive la rețea prin linie comutată, utilizând unul din protocoalele SLIP (Serial Line IP) sau PPP (Point-to-Point Protocol). Trebuie să răspundeți cu Yes dacă doriți să utilizați SLIP sau PPP:

Dummy net driver support (CONFIG_DUMMY) [M/n/y/?] **y**

Dacă sistemul are două linii telefonice și două modemi, puteți activa opțiunea de *echilibrare a încărcării pe liniile seriale*, care permite sistemului Linux să utilizeze două linii telefonice pentru a asigura o singură conexiune de capacitate superioară. Desigur, această opțiune are efect numai dacă este activată și la celălalt capăt al liniilor telefonice. În majoritatea cazurilor, puteți răspunde cu No la această întrebare:

EQL (serial line load balancing) support (CONFIG_EQUALIZER) [M/n/y/?] **y**

Următoarea opțiune vă permite să activați PLIP (abrevierea vine de la Parallel Port Internet Protocol), o modalitate de a conecta în rețea punct-la-punct două sisteme Linux conectând porturile paralele ale acestora:

PLIP (parallel port) support (CONFIG_PLIP) [M/n/y/?] **n**

Răspundeți afirmativ la întrebarea următoare dacă doriți să conectați sistemul Linux la un furnizor de servicii Internet (Internet Service Provider, ISP), utilizând protocolul PPP:

PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?] **y**

*
* CCP compressors for PPP are only built as modules.

Următoarea opțiune vă permite activarea suportului pentru SLIP. În prezent majoritatea PC-urilor utilizează protocolul PPP pentru a se conecta la Internet (nu se va întâmpla însă nimic tragic dacă activați și protocolul SLIP chiar dacă doriți să utilizați PPP):

SLIP (serial line) support (CONFIG_SLIP) [M/n/y/?] **y**

Următoarea opțiune se referă la interfețele „packet radio”. Dacă aveți o astfel de interfață, răspundeți afirmativ; în caz contrar, răspundeți cu No:

Radio network interfaces (CONFIG_NET_RADIO) [N/y/?] **n**

Dacă în sistem se găsește o placă Ethernet, răspundeți cu Yes la următoarea întrebare:

Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?] **y**

În continuare, trebuie să selectați cu exactitate modelul plăcii de rețea instalate în sistemul dumneavoastră. Iată spre exemplu cum se poate indica prezența unei plăci mai vechi 3Com 3C503:

3COM cards (CONFIG_NET_VENDOR_3COM) [Y/n/?] **y**

3c501 support (CONFIG_EL1) [M/n/y/?] **n**

3c503 support (CONFIG_EL2) [M/n/y/?] **y**

3c509/3c579 support (CONFIG_EL3) [M/n/y/?] **n**

3c590/3c900 series (592/595/597/900/905) "Vortex/Boomerang" support

(CONFIG_VORTEX) [M/n/y/?] **n**

AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [Y/n/?] **n**

Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [Y/n/?] **n**

Other ISA cards (CONFIG_NET_ISA) [Y/n/?] **n**

EISA, VLB, PCI and on board controllers (CONFIG_NET_EISA) [Y/n/?] **n**

Pocket and portable adaptors (CONFIG_NET_POCKET) [Y/n/?] **n**

Token ring driver support (CONFIG_TR) [Y/n/?] **n**

FDDI driver support (CONFIG_FDDI) [N/y/?] **n**

ARCnet support (CONFIG_ARCNET) [M/n/y/?] **n**

Următoarea opțiune vă permite să includeți suport pentru ISDN (Integrated Services Digital Network), o linie telefonică digitală cu ajutorul căreia puteți conecta sistemul Linux la Internet. Răspundeți afirmativ numai dacă ați instalat deja serviciile ISDN:

*

* ISDN subsystem

*

ISDN support (CONFIG_ISDN) [M/n/y/?] **n**

Dacă tipul CD-ROM-ului dumneavoastră este comun (IDE sau SCSI), răspundeți cu No la următoarea întrebare:

*

* CD-ROM drivers (not the SCSI or IDE/ATAPI drives)

*

Support for non-SCSI/IDE/ATAPI CD-ROM drives

(CONFIG_CD_NO_IDESCSI) [Y/n/?] **n**

Următorul set de opțiuni permite să activați suportul pentru diferite tipuri de sisteme de fișiere. Puteți răspunde la aceste întrebări consultând indicațiile din tabelul următor:

Opțiune	Descriere
CONFIG_QUOTA	Activează opțiunea „disk quota”, care permite limitarea spațiului pe disc al fiecărui utilizator. Răspundeți cu y numai dacă sistemul are un număr mare de utilizatori și doriți să introduceți o astfel de limitare.
CONFIG_MINIX_FS	Permite utilizarea sistemului de fișiere Minix – sistemul de fișiere inițial utilizat în Linux. Acest sistem simplu de fișiere este încă întrebuințat pe dischete; în concluzie, se recomandă să răspundeți cu m pentru a include suportul pentru acest sistem de fișiere sub formă de modul.
CONFIG_EXT_FS	Activează suportul pentru sistemul extins de fișiere, care nu se mai utilizează. Puteți răspunde cu n la această întrebare.
CONFIG_EXT2_FS	Activează suportul pentru al doilea sistem extins de fișiere – sistemul standard utilizat în prezent în Linux. Trebuie să răspundeți cu y la această întrebare.
CONFIG_XIA_FS	Permite utilizarea sistemului de fișiere XIA, un alt sistem de fișiere care nu mai este utilizat în prezent. Selectați n pentru a dezactiva această opțiune.
CONFIG_FAT_FS	Activează suportul pentru orice sistem de fișiere bazat pe tabele de alocare a fișierelor (FAT); exemple de astfel de sisteme sunt MS-DOS și sistemul VFAT din Windows 95. Răspundeți cu y dacă doriți să aveți acces la fișiere MS-DOS sau VFAT.
CONFIG_MS-DOS_FS	Permite utilizarea sistemului de fișiere MS-DOS. Apăsați y dacă doriți să puteți citi o partiție MS-DOS de pe disc sau de pe o dischetă MS-DOS.
CONFIG_VFAT_FS	Permite utilizarea sistemului de fișiere VFAT din Windows 95. Răspundeți cu y dacă în sistem există partiții cu sistem de fișiere VFAT.
CONFIG_UMSDOS_FS	Permite utilizarea sistemului de fișiere UMSDOS. Acesta permite memorarea fișierelor Linux pe o partiție MS-DOS. Se recomandă să răspundeți cu n .
CONFIG_PROC_PS	Activează suportul pentru sistemul virtual de fișiere /proc, prin intermediul căruia puteți obține informații despre nucleul sistemului de operare. Sistemul de fișiere /proc nu există în mod fizic pe disc; fișierele de aici sunt create în momentul în care sunt utilizate efectiv. Trebuie să răspundeți cu y pentru a activa această opțiune, deoarece unele programe ale sistemului au nevoie de sistemul /proc.

Opțiune	Descriere
CONFIG_NFS_FS	Permite montarea unor directoare NFS (Network File System) aflate pe alte sisteme din rețea. Răspundeți cu y dacă doriți să utilizați directoare NFS.
CONFIG_ROOT_NFS	Permite ca sistemul Linux să monteze întregul sistem de fișiere din alt sistem care rulează NFS (opțiunea este utilă dacă sistemul Linux nu are un hard-disc). În majoritatea cazurilor, răspunsul selectat este n .
CONFIG_SMB_FS	Permite sistemului Linux să aibă acces la directoare partajate de pe PC-urile din rețea care rulează Windows 95 sau Windows for Workgroups. Răspundeți cu y dacă doriți să aveți acces la directoare partajate aflate pe sistemele Windows din rețea.
CONFIG_SMB_WIN95	Permite sistemului Linux să corecteze anumite erori prezente în directoarele Windows 95 partajate. Răspundeți cu y dacă ați activat opțiunea CONFIG_SMB_FS.
CONFIG_ISO9660_FS	Activează suportul pentru sistemul standard ISO9660, utilizat pe CD-ROM-uri (acest sistem a fost cunoscut și sub denumirea de High Sierra File System, fiind referit ca hsts pe anumite stații UNIX). Dacă dispuneți de o unitate CD-ROM, răspundeți cu y .
CONFIG_HPFS_FS	Permite sistemului Linux să aibă acces la un sistem de fișiere OS/2 HPFS. Răspundeți cu y dacă pe disc se găsește cel puțin o partiție OS/2.
CONFIG_SYSV_FS	Activează suportul pentru sistemul de fișiere System V utilizat de variantele de UNIX SCO, Xenix și Coherent, utilizate de asemenea pe calculatoarele Intel. Dacă aveți nevoie de accesul la un sistem de fișiere System V, răspundeți cu y .
CONFIG_UFS_FS	Permite sistemului Linux să citească (dar nu să și scrie) pe un sistem de fișiere UFS, utilizat de variantele de UNIX compatibile BSD (Berkeley Software Distribution); câteva dintre acestea sunt SunOS, FreeBSD, NetBSD sau NeXTstep. Răspundeți cu y dacă trebuie să aveți acces la un sistem de fișiere UFS.

Listingul următor prezintă o posibilitate de configurare a opțiunilor referitoare la sistemele de fișiere:

```
*
* Filesystems
*
Quota support (CONFIG_QUOTA) [y/n/?] n
Minix fs support (CONFIG_MINIX_FS) [M/n/y/?] m
Extended fs support (CONFIG_EXT_FS) [M/n/y/?] n
Second extended fs support (CONFIG_EXT2_FS) [Y/m/n/?] y
xiafs filesystem support (CONFIG_XIA_FS) [M/n/y/?] n
DOS FAT fs support (CONFIG_FAT_FS) [y/m/n/?] y
MSDOS fs support (CONFIG_MSDOS_FS) [y/m/n/?] y
VFAT (Windows-95) fs support (CONFIG_VFAT_FS) [M/n/y/?] n
umsdos: Unix like fs on top of standard MSDOS FAT fs (CONFIG_UMSDOS_FS)
[M/n/y/?] n
/proc filesystem support (CONFIG_PROC_FS) [Y/n/?] y
NFS filesystem support (CONFIG_NFS_FS) [M/n/y/?] y
Root file system on NFS (CONFIG_ROOT_NFS) [N/y/?] (NEW) n
SMB filesystem support (to mount WfW shares etc.) (CONFIG_SMB_FS) [M/n/y/?] y
SMB Win95 bug work-around (CONFIG_SMB_WIN95) [N/y/?] y
ISO 9660 CDRom filesystem support (CONFIG_ISO9660_FS) [M/n/y/?] y
OS/2 HPFS filesystem support (read only) (CONFIG_HPFS_FS) [M/n/y/?] n
System V and Coherent filesystem support (CONFIG_SYSV_FS) [M/n/y/?] n
UFS filesystem support (read only) (CONFIG_UFS_FS) [M/n/y/?] n
```

La următoarea întrebare se răspunde de regulă cu **Yes**, deoarece reprezintă o opțiune necesară ori de câte ori utilizați portul serial al PC-ului (cum ar fi atunci când doriți să conectați un modem):

```
*
* Character devices
```

```
Standard/generic serial support (CONFIG_SERIAL) [Y/m/n/?] y
```

Următoarele câteva opțiuni sunt necesare dacă dispuneți de plăci seriale multiport, care vă permit să conectați terminale multiple sau alte dispozitive la sistemul Linux. Răspundeți cu **No** dacă nu aveți astfel de dispozitive în sistem. Între aceste opțiuni referitoare la dispozitivele seriale multiport se găsește și una despre suportul pentru o imprimantă conectată la portul paralel (CONFIG_PRINTER). Dacă intenționați să conectați o imprimantă la portul paralel, răspundeți cu **Yes** la această opțiune.

```
Digiboard PC/Xx Support (CONFIG_DIGI) [N/y/?] n
Cyclades async mux support (CONFIG_CYCLADES) [M/n/y/?] n
Stallion multiport serial support (CONFIG_STALDRV) [Y/n/?] n
SDL RISCom/8 card support (CONFIG_RISCOM8) [M/n/y/?] n
Parallel printer support (CONFIG_PRINTER) [M/n/y/?] y
Specialix IO8+ card support (CONFIG_SPECIALIX) [M/n/y/?] n
```

Dacă aveți un mouse conectat direct la magistrală sau la portul PS/2, răspundeți cu **Yes** la următoarea opțiune:

```
Mouse support (not serial mice) (CONFIG_MOUSE) [Y/n/?] y
```

Răspundeți apoi cu **Yes** la opțiunea corespunzătoare tipului de mouse pe care îl aveți instalat în sistem:

```
ATIXL busmouse support (CONFIG_ATIXL_BUSMOUSE) [Y/m/n/?] n
Logitech busmouse support (CONFIG_BUSMOUSE) [Y/m/n/?] n
Microsoft busmouse support (CONFIG_MS_BUSMOUSE) [Y/m/n/?] m
PS/2 mouse (aka "auxiliary device") support (CONFIG_PSMOUSE) [Y/m/n/?] y
C&T 82C710 mouse port support (as on TI Travelmate) (CONFIG_82C710_MOUSE) [Y/n/?] n
```

Dacă doriți să utilizați și alte dispozitive, cum ar fi creioanele optice sau ecranele sensibile la atingere, răspundeți afirmativ la următoarea întrebare:

```
Support for user misc device modules (CONFIG_UMISC) [N/y/?] n
```

Dacă unitatea de bandă de care dispuneți nu este SCSI, trebuie să răspundeți cu **Yes** la următoarea întrebare; în caz contrar, răspundeți cu **No**:

```
QIC-02 tape support (CONFIG_QIC02_TAPE) [N/y/?] n
```

Dacă unitatea de bandă este conectată la controllerul unității de dischetă, răspundeți cu **y** sau **m** la opțiunea următoare (luați aminte la comentariul referitor la fixarea valorilor anumitor parametri într-un anumit fișier sursă):

```
Ftape (QIC-80/Travan) support (CONFIG_FTAPE) [M/n/y/?] m
```

```
*
* Set IObase/IRQ/DMA for ftape in .drivers/char/ftape/Makefile
```

Următoarea opțiune se referă la calculatoarele laptop alimentate cu baterii, pentru care economisirea energiei este extrem de importantă. Dacă nu instalați sistemul Linux pe un calculator laptop, răspundeți cu **No** la întrebarea următoare:

```
Advanced Power Management BIOS support (CONFIG_APM) [N/y/?] n
```

Următoarea opțiune permite activarea unui dispozitiv de semnalizare – dacă după ce ai creat un anumit fișier, acesta nu este actualizat cel puțin o dată în fiecare minut, sistemul repornește. De regulă, această opțiune nu este necesară; prin urmare, se recomandă să răspunzi cu No:

Watchdog Timer Support (CONFIG_WATCHDOG) [N/y/?] n

Dacă activezi următoarea opțiune, vei avea acces la ceasul de timp real al PC-ului dumneavoastră prin intermediul unui fișier dispozitiv special. Dacă rulezi o versiune multiprocesor (Symmetric Multi Processing) a sistemului Linux, trebuie să activezi această opțiune; în caz contrar, răspunzi cu No:

Enhanced Real Time Clock Support (CONFIG_RTC) [Y/n/?] n

Următorul set de opțiuni este destinat utilizării plăcii de sunet. Dacă în sistem este instalată o placă de sunet, răspunzi mai întâi cu y sau m la opțiunea CONFIG_SOUND. În continuare, trebuie să răspunzi la un număr de întrebări referitoare la diferite plăci de sunet. Se recomandă să răspunzi cu m, pentru a construi suportul pentru placa de sunet sub forma unor module care pot fi încărcate atunci când sunt necesare. Iată o listă a opțiunilor legate de redarea și prelucrarea sunetului:

```
*
* Sound
*
Sound card support (CONFIG_SOUND) [M/n/y/?] m
/dev/dsp and /dev/audio support (CONFIG_AUDIO) [Y/n/?] y
MIDI interface support (CONFIG_MIDI) [Y/n/?] y
FM synthesizer (YM3812/OPL-3) support (CONFIG_YM3812) [M/n/?] m
ProAudioSpectrum 16 support (CONFIG_PAS) [M/n/?] n
Sound Blaster (SB, SBPro, SB16, clones) support (CONFIG_SB) [M/n/?] n
Generic OPL2/OPL3 FM synthesizer support (CONFIG_ADLIB) [M/n/?] n
Gravis Ultrasound support (CONFIG_GUS) [M/n/?] n
PSS (ECHO-ADI2111) support (CONFIG_PSS) [M/n/?] n
MPU-401 support (NOT for SB16) (CONFIG_MPU401) [M/n/?] n
6850 UART Midi support (CONFIG_UART6850) [M/n/?] n
MPU-401 UART Midi support (CONFIG_UART401) [M/n/?] n
Microsoft Sound System support (CONFIG_MSS) [M/n/?] n
Ensoniq SoundScape support (CONFIG_SScape) [M/n/?] n
MediaTrix AudioTrix Pro support (CONFIG_TRIX) [M/n/?] n
Support for MAD16 and/or Mozart based cards (CONFIG_MAD16) [M/n/?] n
Support for Turtle Beach Wave Front (Maui, Tropez) synthesizers (CONFIG_MAUI)
[M/n/?] n
```

```
Audio DMA buffer size 4096, 16384, 32768 or 65536 (DSP_BUFFSIZE) [65536]
make[1]: Entering directory `/usr/src/linux-2.0.31/drivers/sound'
Compiling Sound Driver v 2 for Linux
rm -f configure
gcc -I/usr/src/linux-2.0.31/include -o configure configure.c
./configure fixedlocal > local.h
./configure fixeddefines > .defines
make[1]: Leaving directory `/usr/src/linux-2.0.31/drivers/sound'
Additional low level drivers (CONFIG_LOWLEVEL_SOUND) [N/y/?] n
```

Următoarea și totodată ultima opțiune vă permite să puteți afla cât timp utilizează nucleul pentru a executa anumite activități. Majoritatea utilizatorilor răspund cu No, pentru a dezactiva această opțiune:

```
*
* Kernel hacking
*
Kernel profiling support (CONFIG_PROFILE) [N/y/?] n
```

The linux kernel is now hopefully configured for your setup.

Check the top-level Makefile for additional configuration, and do a 'make dep; make clean' if you want to be sure all the files are correctly re-made.

Sesiunea de configurare a nucleului se încheie cu afișarea acestui mesaj, care vă informează că pentru reconstruirea corectă a nucleului este necesar să continuați cu secvența de comenzi:

make dep; make clean

Înainte de aceasta, puteți vizualiza și modifica fișierul Makefile, pentru a vă asigura că toate opțiunile au fost corect configurate.

Generarea nucleului

În continuare, vă recomandăm să inițiați următoarele trei activități utilizând o singură linie de comandă (sistemul Linux permite introducerea mai multor comenzi pe o aceeași linie, separate prin caracterul punct și virgulă ';'); prin urmare, este suficient să introduceți linia de comandă, după care apăsați Enter și luați o pauză, deoarece activitatea durează o anumită perioadă.

În funcție de configurația sistemului dumneavoastră, construirea unui nou nucleu poate lua de la câteva minute la mai mult de o oră. Introduceți următoarea linie de comandă pentru a lansa procesul de construire:

make dep; make clean; make boot

Comanda make dep determină fișierele care s-au modificat și trebuie recompilate. Comanda make clean șterge apoi fișierele vechi care nu mai sunt necesare (de exemplu, vechi copii ale nucleului). În fine, make boot va crea noua versiune a nucleului.

Pe măsură ce nucleul este construit, pe ecran apare o mulțime de mesaje. În final, noul nucleu se va găsi sub forma unui fișier comprimat numit zImage din directorul /usr/src/linux/arch/i386/boot.

Pentru a utiliza efectiv noul nucleu, trebuie să copiați nucleul în directorul /boot și să editați fișierul /etc/lilo.conf pentru a configura programul LILO – cel care încarcă sistemul Linux. Acești pași sunt prezentați în secțiunea „Instalarea unei versiuni a nucleului”. Înainte de a trece însă la instalarea nucleului, este necesar să construiți și să instalați modulele necesare.

Generarea și instalarea modulelor

Dacă ați selectat anumite module atunci când ați configurat nucleul, trebuie să construiți aceste module și apoi să le instalați. Efectuați aceste activități urmând etapele:

1. Introduceți următoarele comenzi pentru a construi modulele:

```
cd /usr/src/linux
make modules
```

2. Setul curent de module se găsește într-un director al cărui nume coincide cu versiunea curentă a nucleului pe care sistemul îl rulează în mod curent. De exemplu, dacă sistemul rulează un nucleu versiunea 2.0.31, modulele se găsesc în directorul:

```
/lib/modules/2.0.31
```

Schimbați numele directorului care conține modulele, după cum urmează:

```
mv /lib/modules/2.0.31 /lib/modules/2.0.31-old
```

3. Instalați noile module utilizând comanda:

```
make modules_install
```


Puteți acum instala și noul nucleu, după care, utilizând LILO, puteți încerca să îl utilizați pentru pornirea sistemului.

Instalarea unei noi versiuni a nucleului

Sistemul Red Hat Linux utilizează programul LILO pentru a încărca nucleul sistemului de pe disc. Fișierul de configurare `/etc/lilo.conf` indică fișierul binar pe care programul LILO îl încarcă. Puteți examina conținutul fișierului de configurare introducând comanda:

```
cat /etc/lilo.conf
```

Iată ce se poate observa atunci când executăm această comandă pe un sistem care dispune de un adaptor SCSI:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.31
    label=linux
    root=/dev/hda1
    initrd=/boot/initrd-2.0.31.img
    read-only
```

În capitolul 11 puteți afla mai multe detalii despre LILO și fișierul său de configurare (`/etc/lilo.conf`). Acum este suficient să rețineți următoarele:

- Ultimele cinci linii definesc un anumit nucleu, pe care LILO îl poate încărca. Puteți modifica nucleul pe care LILO îl încarcă adăugând o secțiune similară de cinci linii în fișierul de configurare.
- Linia `image=/boot/vmlinuz-2.0.31` identifică efectiv nucleul pe care LILO îl încarcă. În acest caz, nucleul este `vmlinuz-2.0.31` și se găsește în directorul `/boot`.
- Linia `label=linux` asociază nucleului un nume. În acest caz, este suficient să tastați `linux` la promptul `boot`: și LILO va încărca acest nucleu.
- Linia `root=/dev/hda1` precizează partiția discului pe care se găsește rădăcina sistemului de fișiere. În sistemul dumneavoastră, este posibil ca aceasta să se afle pe o partiție diferită.
- * Linia `initrd=/boot/initrd-2.0.31.img` precizează fișierul care conține imaginea discului RAM inițial, care este utilizat ca sistem de fișiere înainte ca hard-discul să fie disponibil. Linia `initrd` apare numai dacă sistemul dispune de un adaptor SCSI, iar nucleul utilizează un driver SCSI modular. În acest caz, nucleul utilizează discul RAM – un bloc de memorie utilizat ca disc – pentru a porni inițial; ulterior, driverul modular SCSI este încărcat și nucleul va utiliza discul SCSI. Nu aveți nevoie de linia `initrd` în cazul în care suportul pentru adaptorul SCSI este încorporat în nucleu.

Pe sistemele care au partiții MS-DOS, fișierul de configurare pentru LILO poate conține o altă secțiune, cu detalii despre sistemul de operare (de cele mai multe ori Windows 95) instalat pe acea partiție.

Pentru a configura LILO astfel încât să poată încărca un alt nucleu (cel pe care l-ați construit chiar acum), parcurgeți următoarele etape:

1. Copiați fișierul binar care reprezintă noul nucleu în directorul `/boot`. Noul fișier binar se numește `/usr/src/linux/arch/i386/boot/zImage`. Este suficient să îl copiem în directorul `/boot`, fără a-i modifica numele:

```
cp /usr/src/linux/arch/i386/boot/zImage /boot
```

La alegere, puteți da noului fișier un nume diferit (ceea ce este absolut necesar atunci când un fișier mai vechi numit tot `zImage` se găsește deja în acel director).

2. Utilizați editorul preferat de texte pentru a modifica fișierul `/etc/lilo.conf`, adăugând următoarele linii imediat după linia care precizează perioada de întârziere (dacă nu cunoașteți nici un editor de texte, consultați capitolul 24 pentru a învăța cum puteți utiliza `vi` sau `Emacs` – două dintre cele mai cunoscute editoare din Linux):

```
image=/boot/zImage
    label=new
    root=/dev/hda1
    read-only
```

În sistemul dumneavoastră, aveți grijă ca linia `root` să fie corectă – înlocuiți dacă este cazul `/dev/hda1` cu partiția discului în care se află directorul rădăcină (`/`) al sistemului Linux.

Observați că am omis linia cu `initrd`, presupunând că nu utilizați un driver SCSI modular, chiar dacă sistemul dispune de un adaptor SCSI.

3. Salvați fișierul `lilo.conf` și părăsiți editorul de texte.

4. Reinstalați LILO utilizând comanda:

```
/sbin/lilo
```

În acest moment, puteți reporni sistemul pentru a testa noul nucleu.

Repornirea sistemului

După ce ați terminat de configurat și de instalat programul LILO și ați revenit la promptul sistemului Linux, tastați următoarea comandă pentru a reporni sistemul:

```
reboot
```

La promptul `boot` :, tastați numele pe care l-ați asociat noului nucleu în fișierul `/etc/lilo.conf`. Dacă ați adăugat descrierea noului nucleu înaintea celorlalte sisteme de operare în fișierul de configurare, nu mai trebuie să tastați nimic.

După repornirea sistemului, trebuie să ajungeți la promptul binecunoscut `login`. Pentru a vă asigura că rulați într-adevăr noul nucleu, intrați în sistem ca utilizator obișnuit și tastați:

```
uname -srv
Linux 2.0.31 #4 Fri Apr 10 21:17:07 EDT 1998
```

Ultima parte a mesajului vă arată data și ora la care nucleul curent a fost construit – acestea trebuie să coincidă cu data și ora la care ați reconstruit nucleul.



Sugestie

Dacă sistemul se blochează (adică se pare că nu se întâmplă nimic – pe ecran nu se afișează nimic, iar discul nu are activitate), este posibil să fi omis un pas pe parcursul operației de reconstrucție. Puteți opri alimentarea PC-ului, repornindu-l după aceasta. De această dată, introduceți `linux` (numele vechiului nucleu, care funcționează corect) la promptul `boot` : al programului LILO.

Dacă nu puteți încărca nici versiunea anterioară a sistemului Linux, utilizați o dischetă de pornire în caz de urgență (care conține o versiune anterioară funcțională a sistemului Linux) pentru a reporni sistemul. Puteți acum să repetați construirea nucleului și procesul de instalare, asigurându-vă că efectuați corect toate activitățile.

Actualizarea nucleului utilizând un fișier RPM

După cum am amintit în secțiunea „Actualizarea nucleului prin aplicarea unor patch-uri”, este posibil ca la aplicarea patch-urilor să apară erori cauzate de diferențele dintre versiunea fișierelor nucleului și versiunea surselor standard ale nucleului. În acest caz, puteți obține versiunile mai noi ale nucleului sistemului direct de pe serverul FTP al firmei Red Hat. Firma Red Hat asigură distribuirea tuturor noilor versiuni de software, inclusiv a noilor versiuni ale nucleului sistemului Linux, sub formă de fișiere RPM.

Etapele principale ale operației de actualizare a nucleului sunt în acest caz următoarele:

1. Obțineți fișierul RPM care reprezintă nucleul Red Hat de pe serverul FTP al firmei Red Hat. Este de asemenea necesar să obțineți și fișierul RPM `kernel-modules`. În plus, dacă doriți să reconstruiți nucleul, trebuie să transferați și fișierele `kernel-source` și `kernel-headers` corespunzătoare noii versiuni a nucleului.
 2. Instalați fișierele RPM utilizând comanda `rpm -i`.
 3. Dacă aveți un adaptor SCSI în configurația sistemului, creați un nou disc RAM inițial executând comanda `/sbin/mkinitrd`.
 4. Reconfigurați programul LILO pentru a încărca noua versiune a nucleului.
 5. Încercați noua versiune repornind sistemul.
- Acești pași sunt prezentați pe larg în următoarele secțiuni.

Transferarea noilor fișiere RPM

Firma Red Hat pune la dispoziție toate noile versiuni de software sub formă de fișiere RPM – pachete – prin intermediul serverului FTP propriu, `ftp.redhat.com`. Pentru versiunea Red Hat Linux 5.0, aceste noi versiuni se găsesc în directorul `/pub/redhat/updates/5.0/i386`. Pentru versiunea Red Hat Linux 5.1, noile versiuni se găsesc în directorul `/pub/redhat/updates/5.1/i386`.

Sesiunea FTP prezentată în continuare arată cum se pot transfera de pe serverul firmei Red Hat fișierele RPM necesare efectuării unei actualizări a nucleului (comenzile introduse de la tastatură apar scrise îngroșat):

```
ftp ftp.redhat.com
Connected to ftp.redhat.com.
220-
220-           Red Hat Software Proudly Presents:
220-
220-           Red Hat 5.0 – Hurricane,
ftp://ftp.redhat.com/pub/redhat/redhat-5.0/
220-           Red Hat Contrib – ftp://ftp.redhat.com/pub/contrib/
220-
220 gonzales.redhat.com FTP server (Version wu-2.4.2-academ[BETA-15])(1) Fri Dec
12 20:41:30 EST 1997) ready.
Name (ftp.redhat.com:naba): anonymous
331 Guest login ok, send your complete e-mail address as Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/redhat/updates/5.0/i386
250- Please read the file README.kernl
250- it was modified on Wed Dec 17 14:05:51 1997 – 109 days ago
250 CWD command successful.
```

```
ftp> ls kernel*
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
-rw-r--r-- 1 root root 482029 Nov 26 11:31 kernel-2.0.32-2.i386.rpm
-rw-r--r-- 1 root root 446095 Nov 26 11:31 kernel-headers-2.0.32-2.i386.rpm
-rw-r--r-- 1 root root 1229707 Nov 26 11:31 kernel-modules-2.0.32-2.i386.rpm
-rw-r--r-- 1 root root 6195859 Nov 26 11:31 kernel-source-2.0.32-2.i386.rpm
226 Transfer complete.
ftp> binary
200 Type set to I.
ftp> prompt
Interactive mode off.
ftp> mget kernel*
local: kernel-2.0.32-2.i386.rpm remote: kernel-2.0.32-2.i386.rpm
200 PORT command successful.
150 Opening BINARY mode data connection for kernel-2.0.32-2.i386.rpm (482029
bytes).
226 Transfer complete.
482029 bytes received in 240 secs (2 Kbytes/sec)
local: kernel-headers-2.0.32-2.i386.rpm remote: kernel-headers-2.0.32-2.i386.rpm
200 PORT command successful.
150 Opening BINARY mode data connection for kernel-headers-2.0.32-2.i386.rpm
(446095 bytes).
226 Transfer complete.
446095 bytes received in 260 secs (1.7 Kbytes/sec)
local: kernel-modules-2.0.32-2.i386.rpm remote: kernel-modules-2.0.32-2.i386.rpm
200 PORT command successful.
150 Opening BINARY mode data connection for kernel-modules-2.0.32-2.i386.rpm
(1229707 bytes).
226 Transfer complete.
1229707 bytes received in 565 secs (2.1 Kbytes/sec)
local: kernel-source-2.0.32-2.i386.rpm remote: kernel-source-2.0.32-2.i386.rpm
200 PORT command successful.
150 Opening BINARY mode data connection for kernel-source-2.0.32-2.i386.rpm
(6195859 bytes).
226 Transfer complete.
6195859 bytes received in 2.35e+03 secs (2.6 Kbytes/sec)
```

În sesiunea precedentă, am transferat următoarele fișiere RPM:

Numele fișierului RPM	Descriere
<code>kernel-2.0.32-2.i386.rpm</code>	Include nucleul pentru versiunea 2.0.32
<code>kernel-modules-2.0.32-2.i386.rpm</code>	Include toate modulele necesare în versiunea 2.0.32
<code>kernel-headers-2.0.32-2.i386.rpm</code>	Conține fișierele antet pentru versiunea 2.0.32 (necesare dacă doriți să reconstruiți nucleul)
<code>kernel-source-2.0.32-2.i386.rpm</code>	Conține fișierele sursă pentru versiunea 2.0.32 a sistemului Linux (necesare dacă doriți să reconstruiți nucleul)

Instalarea fișierelor RPM

Pentru a instala nucleul și modulele necesare, parcurgeți următoarele etape:

1. Asigurați-vă că directorul curent de lucru este chiar directorul în care se găsesc fișierele RPM (cele pe care tocmai le-ați transferat de pe serverul FTP al firmei Red Hat).
2. Tastați următoarea comandă, pentru a instala nucleul din fișierul RPM corespunzător:

```
rpm -ivh kernel-2.0.32-2.i386.rpm
kernel
```

```
#####
```

Acest pas determină crearea noului fișier `vmlinux-2.0.32` în directorul `/boot`.

3. Instalați noile module ale nucleului, executând comanda:

```
rpm -ivh kernel-modules-2.0.32-2.i386.rpm
kernel-modules
#####
```

Acest pas determină crearea directorului `/lib/modules/2.0.32` și plasează modulele necesare nucleului în acest director.

Fișierele RPM `kernel-source` și `kernel-headers` trebuie instalate numai dacă doriți să construiți un nou nucleu. Utilizați următoarele comenzi `rpm -i` pentru instalarea acestor pachete:

```
rpm -i kernel-source-2.0.32-2.i386.rpm
rpm -i kernel-headers-2.0.32-2.i386.rpm
```

Crearea unui disc RAM inițial

Dacă în configurație aveți un adaptor SCSI și driverul pentru acesta este încărcat ca modul, trebuie să creați un nou *disc RAM inițial*; acesta este un fișier care poate fi copiat într-un bloc din memorie și apoi utilizat ca disc rezident în memorie.

Pentru a stabili dacă driverul pentru adaptorul SCSI este modular, utilizați comanda următoare:

```
grep scsi /etc/conf.modules
```



Sugestie

Dacă această comandă nu afișează nimic, înseamnă că sistemul dumneavoastră nu utilizează module SCSI și prin urmare nu necesită un disc RAM inițial. Puteți trece peste acest pas, continuând cu reconfigurarea utilitarului LILO.

Spre exemplu, pe un sistem cu un driver SCSI Adaptec AHA 1542, putem obține ca rezultat al comenzii `grep`:

```
grep scsi /etc/conf.modules
alias scsi_hostadapter aha1542
```

În acest caz, crearea discului inițial s-a efectuat utilizând comanda `mkinitrd`. De obicei, discul RAM inițial este memorat într-un fișier al cărui nume începe cu `initrd`. După cum v-ați dat seama, `initrd` reprezintă o prescurtare de la „initial RAM disk” (disc RAM inițial), iar comanda `mkinitrd` este astfel numită deoarece creează un fișier `initrd`.

Pentru a crea discul RAM inițial pentru versiunea 2.0.32 a nucleului, tastați comanda:

```
/sbin/mkinitrd /boot/initrd-2.0.32.img 2.0.32
```

Ca rezultat, se va crea în directorul `/boot` fișierul `initrd-2.0.32.img`. Acest fișier `initrd` va fi referit în fișierul de configurare al utilitarului LILO (`/etc/lilo.conf`).

Reconfigurarea utilitarului LILO

După ce ați instalat fișierele RPM și, dacă este cazul, ați creat și discul RAM inițial, trebuie să reconfigurați utilitarul LILO pentru a putea să încărcați noul nucleu. Acest pas de reconfigurare este asemănător cu procesul descris anterior în secțiunea „Instalarea unui nou nucleu”. Singura diferență constă în obligativitatea de a include o linie `initrd` dacă sistemul necesită un driver SCSI modular.

Utilizați un editor de texte cum este `vi` sau `Emacs` pentru a modifica fișierul `/etc/lilo.conf`. Adăugați următoarele linii la începutul fișierului, imediat după linia `timeout`:

```
image=/boot/vmlinux-2.0.32
label=linux-new
root=/dev/hda1
initrd=/boot/initrd-2.0.32.img
read-only
```



Atenție!

În linia `root=/dev/hda1`, modificați `/dev/hda1` cu numele exact al dispozitivului corespunzător partiției pe care se găsește directorul rădăcină (`/`) al sistemului Linux. După ce ați editat și ați salvat fișierul `/etc/lilo.conf`, reinstalați utilitarul LILO cu ajutorul următoarei comenzi:

```
/sbin/lilo
```

În acest moment, puteți reporni sistemul pentru a testa noua versiune a nucleului.

Testarea unui nou nucleu

Sunteți pe punctul de a testa noul nucleu după ce ați instalat fișierele RPM `kernel` și `kernel-modules`, ați creat fișierul care conține discul RAM inițial și ați reconfigurat utilitarul LILO. De la promptul sistemului Linux, introduceți comanda:

```
reboot
```

După ce sistemul repornește, la promptul `boot:`, introduceți `linux-new` pentru a încărca noua versiune a nucleului. Dacă ați adăugat descrierea pentru noul nucleu înaintea tuturor celorlalte sisteme de operare în fișierul de configurare al utilitarului LILO, noul nucleu va fi încărcat chiar dacă nu introduceți nimic la promptul `boot:`.

După ce sistemul Linux pornește, înaintea afișării promptului `login:`, trebuie să apară următorul mesaj:

```
Red Hat Linux release 5.0 (Hurricane)
Kernel 2.0.32 on an i586
```

În acest moment, sistemul dumneavoastră rulează un nucleu cu versiunea 2.0.32.

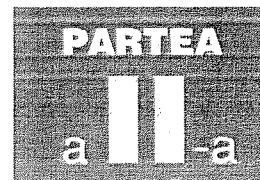
Rezumat

Sistemul Linux s-a dezvoltat ca urmare a efortului depus de mii de programatori din întreaga lume. Acești programatori (și mulți alții) continuă să îmbunătățească sistemul Linux, lansând în permanență corectări ale unor erori sau noi versiuni. Se recomandă ca în mod periodic să transferați din Internet unele dintre aceste noi versiuni sau corectări și să reconstruiți nucleul sistemului. În acest capitol, se găsesc informații detaliate referitoare la modul în care se pot aplica aceste corectări (denumite *patch-uri*) și la operația de construire a unui nou nucleu. Tot de aici puteți afla cum să transferați și cum să instalați noile versiuni puse la dispoziție de firma Red Hat sub forma fișierelor Red Hat Package Manager (RPM). Următorul capitol trece de la operațiile de instalare și configurare la evidențierea principalelor caracteristici ale sistemului Linux și principalelor activități în care puteți utiliza un PC pe care ați instalat Linux.

După ce ați parcurs acest capitol, ați învățat următoarele:

► Termenul de *nucleu* (*kernel*) se referă la inima sistemului de operare Linux – adică la programul care transformă un PC obișnuit într-un sistem Linux.

- Nucleul Linux este permanent îmbunătățit pentru a corecta anumite erori sau pentru a adăuga noi funcții. Aceste actualizări sunt distribuite sub formă de patch-uri – modificări ale fișierelor sursă Linux. Pentru a beneficia de îmbunătățirile aduse nucleului, trebuie să aplicați aceste patch-uri și apoi să reconstruiți nucleul.
- Patch-urile pot fi obținute prin FTP de pe situl `ftp.funet.fi`, în directorul `/pub/Linux/kernel/src`. Căutați în directorul corespunzător numărului versiunii (spre exemplu, `v2.0` corespunde versiunii 2.0) pentru a obține patch-urile de care aveți nevoie.
- Patch-urile sunt distribuite ca fișiere comprimate – după cum puteți ușor recunoaște datorită prezenței sufixului `.gz` în numele acestora. Trebuie să utilizați comanda `gunzip` pentru a decompresa aceste fișiere.
- Copiați patch-urile în directorul `/usr/src` și apoi utilizați comanda `patch` pentru a le aplica efectiv. În urma aplicării, codul sursă al nucleului din directorul `/usr/src/linux` va fi modificat.
- Înainte de a reconstrui nucleul, pregătiți o dischetă de pornire în caz de urgență pentru a putea reporni sistemul dacă procesul de reconstruire nu decurge normal.
- Pentru a reconstrui nucleul, deplasați-vă în directorul `/usr/src/linux` și utilizați comanda `make`. Secvența de comenzi `make` pe care trebuie să o introduceți este (în ordine) `make config`; `make dep`; `make clean`; `make boot`. Trebuie de asemenea să introduceți comenzile `make modules` și `make modules_install`.
- Pentru ca noul nucleu să poată fi încărcat, trebuie mai întâi copiat în directorul `/boot`, după care fișierul `/etc/lilo.conf` trebuie editat pentru a include informații referitoare la noul nucleu. Trebuie să introduceți apoi comanda `/sbin/lilo` pentru a reinstala utilitarul LILO.
- O altă soluție de actualizare a nucleului (în afară de cea care presupune aplicarea unor patch-uri) este de a transfera și de a instala noile versiuni direct de pe serverul FTP al firmei Red Hat. Aceste noi versiuni sunt distribuite sub formă de fișiere Red Hat Package Manager (RPM), care împachetează toate fișierele necesare unui pachet software într-un singur fișier. Pentru a instala și utiliza fișierele RPM, trebuie să utilizați comanda `rpm`.
- Puteți transfera noile modificări ale sistemului Red Hat Linux de pe serverul `ftp.redhat.com` din directorul `/pub/redhat/updates/5.1/i386`.



RULAREA SISTEMULUI LINUX

- Capitolul 3: O privire generală asupra sistemului Linux
- Capitolul 4: Secretele sistemului X sub Linux
- Capitolul 5: Personalizarea lansării sistemului Linux
- Capitolul 6: Secretele comenzilor Linux
- Capitolul 7: Secretele sistemului DOS sub Linux
- Capitolul 8: Programarea sub Linux în Tcl/Tk

Capitolul 3 O privire generală asupra sistemului Linux

În acest capitol:

- Determinarea versiunii curente a sistemului și semnificația numerelor versiunilor
- Sistemul Linux ca platformă UNIX
- Sistemul X – o interfață grafică pentru Linux
- Facilitățile native de lucru în rețea oferite de Linux
- Utilitatea sistemului Linux pentru administratorii de rețele și de sisteme
- Coexistența dintre sistemele DOS și Linux
- Instrumente pentru dezvoltarea de programe sub Linux
- Sistemul Linux ca poartă spre Internet

După ce ați încheiat cu succes operațiile de instalare și configurare, sunteți pe punctul de a explora și utiliza efectiv sistemul Linux. Probabil că ați vizat deja un anumit scop atunci când v-ați decis să instalați sistemul Linux pe PC-ul dumneavoastră. Indiferent dacă intenționați să utilizați sistemul pentru dezvoltarea de programe sau doar pentru navigarea pe Internet, această carte reprezintă ghidul dumneavoastră personal prin labirintul Linux. Chiar dacă aveți un scop precis, veți găsi utilă o prezentare detaliată a întregului sistem, cu avantajele și dezavantajele pe care acesta le prezintă. Când veți ști mai multe despre Linux, veți putea utiliza sistemul pentru activități la care nici nu v-ați fi gândit înainte.

Acest capitol are ca scop trasarea unui portret general al întregului sistem Linux, descriind modul în care puteți utiliza cât mai eficient majoritatea facilităților native ale sistemului, printre care lucrul în rețea, dezvoltarea de programe și rularea diverselor aplicații.

Versiunile sistemului Linux

După lansarea versiunii 1.0 a sistemului Linux la data de 14 martie 1994, comunitatea care asigura dezvoltarea sistemului a adoptat o schemă de numerotare a versiunilor. Versiunile 1.x.y sau 2.x.y, în care numărul *x* este par, reprezintă versiuni stabile. Numărul *y* se numește *nivel de patch* și este incrementat pe măsură ce anumite probleme sunt rezolvate.

Versiunile 2.x.y, în care numărul *x* este impar, sunt versiuni beta, destinate numai celor care asigură dezvoltarea sistemului. Aceste versiuni pot prezenta o funcționare nesigură; prin urmare, nu se recomandă să le utilizați în activitatea curentă. Cei care asigură dezvoltarea sistemului adaugă noile facilități la aceste versiuni imparate ale sistemului.

În momentul scrierii lucrării de față, ultima versiune stabilă a sistemului Linux era 2.0.34, iar comunitatea care asigură dezvoltarea sistemului lucra asupra versiunii 2.1.1069 (informația privitoare la ultimele versiuni ale nucleului sistemului Linux poate fi găsită pornind de la adresa <http://www.linuxresources.com/what.html>). CD-ROM-ul atașat lucrării conține ultima versiune stabilă a nucleului Linux disponibilă în toamna anului 1998.



Referință încrucișată

Dacă ați aflat de apariția unei versiuni mai noi a sistemului Linux sau a unor patch-uri care vă sunt efectiv utile, transferați aceste patch-uri și reconstruiți nucleul urmând indicațiile din capitolul 2.

Sistemul Linux ca platformă UNIX

Ca și alte sisteme UNIX, Linux este un sistem de operare multiutilizator și multitasking, adică permite conectarea mai multor utilizatori și rularea mai multor programe în mod simultan.

Sistemul Linux a fost proiectat astfel încât să fie compatibil cu standardul IEEE 1003.1/1990 (numit și POSIX.1). Acest standard definește funcțiile pe care aplicațiile scrise în limbajul de programare C le pot utiliza pentru accesul la serviciile sistemului de operare – pentru activități care încep de la deschiderea unui fișier și până la alocarea memoriei. La data de 8 martie 1996, laboratorul de informatică al Institutului Național pentru Standarde și Tehnologii (National Institute of Standards and Technology - NIST) – agenție aflată sub jurisdicția Guvernului Statelor Unite – a stabilit că versiunea 1.2.13 a sistemului Linux distribuită de firma Open Linux Ltd. se conformează standardului POSIX.1. Pentru a consulta o listă a produselor validate de standardul POSIX, încărcați în browserul Web adresa <ftp://nemo.ncsl.nist.gov/pub-posix/151-2reg>.

Pe lângă compatibilitatea cu standardul POSIX, sistemul Linux include caracteristici și din alte standarde UNIX, printre care System V Interface Document (SVID) sau versiunea Berkeley Software Distribution (BSD). Soluția aleasă de Linux este în primul rând pragmatică, fiind orientată pe selectarea celor mai necesare facilități prezente în celelalte tipuri standard de sisteme UNIX.

Pe lângă compatibilitatea cu POSIX.1, sistemul Linux este compatibil și cu POSIX.2 (referitor la shell și utilitare). Standardul POSIX.2 se referă la interpretorul de comenzi al sistemului de operare (cunoscut de regulă sub numele de *shell*) și la un set standard de programe utilitare. Dacă aveți cât de cât o experiență de lucru sub UNIX, știți deja că această familie de sisteme de

operare este orientată spre instrumente. Pentru orice activitate pe care doriți să o desfășurați, sistemul pune la dispoziție un anumit instrument; în plus, shellul permite combinarea mai multor instrumente pentru a executa activități mai complicate decât cele efectuate de instrumentele de bază. Standardul POSIX.2 menține această viziune orientată pe instrumente, stabilind următoarele caracteristici:

- Un shell cu un set specificat de comenzi native și cu o sintaxă bine stabilită, care poate fi utilizat pentru a scrie *programe shell*, numite de regulă scripturi.
- Un set standard de programe utilitare – cum ar fi spre exemplu *sed*, *tr* sau *awk* – care pot fi apelate din programele shell sau direct din aplicații. Setul standard de utilitare include chiar și editorul de texte *vi* sau programul de gestionare a poștei electronice *mail*. Veți afla mai multe despre aceste utilitare în capitolul 6.
- Un set de funcții C, cum ar fi *system* sau *getenv*, pe care aplicațiile le pot utiliza pentru a avea acces la facilitățile oferite de shell.
- Un set de utilitare pentru dezvoltarea de aplicații shell (există un compilator de C compatibil POSIX.2 numit *c89*).

Standardul POSIX

Abrevierea POSIX provine de la Portable Operating System Interface (ajustată astfel încât să rimeze cu UNIX). Institutul Inginerilor Electrotehniști și Electroniști (Institute of Electrical and Electronics Engineers – IEEE) a dezvoltat standardele POSIX pentru a promova portabilitatea aplicațiilor pe diferitele platforme UNIX. Standardele POSIX nu se limitează însă la UNIX. Multe alte sisteme de operare, printre care sistemul VMS al firmei DEC și sistemul Windows NT al firmei Microsoft, implementează standardul POSIX. Acest standard, cunoscut sub denumirea completă de POSIX.1 și care poartă numărul IEEE 1003.1/1990, stabilește interfața de programare a aplicațiilor (Application Programming Interface – API) în limbajul C cu serviciile oferite de sistemul de operare, cum ar fi citirea sau scrierea în fișiere. Standardul POSIX.1 a fost acceptat de Organizația Internațională a Standardelor (International Standard Organization – ISO), fiind înregistrat cu numărul ISO/IEC IS 9945-1:1990.

Prin convenție, termenul POSIX este sinonim cu P1003, întrebuintat de IEEE pentru a denumi activitățile legate de POSIX. Prin urmare, POSIX.1 este sinonim cu P1003.1.

Pe lângă POSIX.1, familia de standarde POSIX include alte 21 de standarde, numerotate de la POSIX.2 până la POSIX.22, după cum urmează:

POSIX.1 – standard acceptat pe scară largă pentru asigurarea portabilității la nivelul codului sursă. POSIX.1 stabilește o interfață de programare a aplicațiilor (API) cu sistemul de operare. Acest standard a fost aprobat de IEEE și ISO. În ultimii ani, POSIX.1 a fost extins pentru a cuprinde și următoarele domenii: POSIX.1a (extensii ale interfeței cu sistemul de operare), POSIX.1b (interfață pentru lucrul în timp real), POSIX.1c (controlul firelor de execuție), POSIX.1d (extensii ale interfeței de lucru în timp real), POSIX.1e (asigurarea securității), POSIX.1f (accesul transparent la fișiere), POSIX.1g (servicii independente de protocol) și POSIX.1h (toleranța la defecte).

POSIX.2 – un standard pentru shell și instrumente, care reprezintă de fapt procesorul de comenzi și programele utilitare pe care un sistem de operare trebuie să le ofere. Acest standard a fost aprobat de IEEE.

POSIX.3 – standard pentru testarea și verificarea compatibilității POSIX. Acest standard a fost aprobat de IEEE.

POSIX.4 – standard pentru programarea în timp real și controlul firelor de execuție, care reprezintă blocuri ale unui același program aflate simultan în execuție. Acest standard propus a fost încorporat în POSIX.1 ca POSIX.1b (interfață pentru lucrul în

(continuare)

(continuare)

timp real), POSIX.1c (controlul firelor de execuție) și POSIX.1d (extensii ale interfeței de lucru în timp real). Standardele POSIX.1b și POSIX.1c au fost aprobate de IEEE.

POSIX.5 – oferă o interfață de programare în limbajul Ada corespunzătoare cu standardul POSIX.1. Acest standard a fost aprobat atât de IEEE, cât și de ISO.

POSIX.6 – standard referitor la securitatea unui sistem. Acest standard a fost redenumit POSIX.1e și se află încă în curs de elaborare.

POSIX.7 – standard privitor la administrarea unui sistem. Acest standard se găsește în prezent în alt domeniu, fiind cunoscut sub numele de P1387.

POSIX.8 – standard care privește interfețele de rețea, incluzând: (a) accesul transparent la fișiere, (b) interfața cu rețeaua independentă de protocol, (c) apelurile de proceduri aflate la distanță (Remote Procedure Calls – RPC) și (d) interfețe dependente de protocol pentru interconectarea sistemelor deschise. Acest standard a fost redenumit POSIX.1f, nefiind aprobat până în prezent.

POSIX.9 – o interfață în limbajul FORTRAN echivalentă cu cea oferită de POSIX.1. Acest standard a fost aprobat atât de IEEE, cât și de ISO.

POSIX.10 – un standard pentru mediul de dezvoltare a aplicațiilor pe supercomputere. Acest standard a fost aprobat de IEEE.

POSIX.11 – standard propus pentru un mediu de dezvoltare a aplicațiilor care prelucrează tranzacții. În prezent, s-a renunțat la acest standard.

POSIX.12 – standard pentru serviciile independente de protocol. Acest standard a fost redenumit POSIX.1g, fiind încă în curs de elaborare.

POSIX.13 – standard pentru un mediu de dezvoltare a aplicațiilor în timp real (aflat în curs de elaborare).

POSIX.14 – standard pentru un mediu de dezvoltare a aplicațiilor distribuite (aflat în curs de elaborare).

POSIX.15 – standard pentru prelucrările orientate pe loturi (aflat în curs de elaborare).

POSIX.16 – standard la care s-a renunțat.

POSIX.17 – standard pentru serviciile X.400 și X.500. Între timp, standardul a fost împărțit și redenumit, alcătuindu-se standardele P1224.2, P1326.2, P1327.2 și P1328.2 (aprobat de către IEEE).

POSIX.18 – reprezintă profilul standardelor POSIX (aflat în curs de elaborare).

POSIX.19 – reprezintă versiunea în limbajul Fortran 90 pentru POSIX.1, dar între timp s-a renunțat la acest standard.

POSIX.20 – standard pentru extensiile pentru lucrul în timp real în limbajul Ada; a fost redenumit POSIX.5b (aflat în curs de elaborare).

POSIX.21 – standard pentru comunicația în timp real într-un sistem distribuit (aflat în curs de elaborare).

POSIX.22 – un cadru de asigurare a securității aplicațiilor (aflat în curs de elaborare).

Pentru informații de ultimă oră privind standardele POSIX, încărcați în browserul Web pagina Comitetului pentru Standarde în domeniul Portabilității Aplicațiilor (Portable Application Standards Committee – PASC), care se găsește la adresa: <http://www.pasc.org> (pagina menționează că este realizată pe un sistem Red Hat Linux cu un server Web Apache).

Shellul prestabilit în Linux este numit *bash*, prescurtare care provine de la *Bourne-Again Shell* – un nume care evocă mai vechiul *shell Bourne*, care a fost shellul prestabilit al sistemelor UNIX încă de la apariția acestora. Shellul *bash* încorporează multe dintre facilitățile cerute de standardul POSIX.2 și altele pe lângă acestea. În mare, *bash* moștenește facilitățile și funcționalitatea oferite de Bourne shell. În cazul unor discrepanțe dintre Bourne shell și POSIX.2, *bash* a adoptat soluția propusă de POSIX.2. Pentru a asigura o compatibilitate și mai strictă cu standardul POSIX.2, *bash* include de asemenea un mod POSIX.

Sistemul Linux constituie un bun exemplu pentru cei care doresc să învețe UNIX, deoarece oferă un set standard de comenzi UNIX (standardul POSIX.2, împreună cu cele mai importante facilități prezente atât în System V, cât și în BSD UNIX).

Compatibilitatea sistemului Linux cu standardul POSIX.1, cât și cele mai importante *apeluri sistem* (funcții apelate în aplicații) îl recomandă ca o excelentă platformă pentru dezvoltarea de programe. Un alt ingredient nelipsit de pe orice stație modernă de lucru – sistemul X Window – este de asemenea prezent și în Linux, într-o variantă numită XFree86.

Disponibilitatea unor programe utilizate în activitatea de zi cu zi – cum ar fi editoarele de texte, foile de calcul sau aplicațiile pentru baze de date – reprezintă până nu demult un punct slab al sistemului Linux. Situația este însă pe punct de a se schimba, produsul Applixware Office Suite (comercializat de firma Red Hat Software) constituind un bun exemplu de software sub Linux utilizabil în activitatea cotidiană. Alte produse recente din aceeași categorie sunt WordPerfect 7 pentru Linux, produs de firma Corel Corporation, respectiv Star Office, distribuit de Star Division (<http://www.stardivision.com/>). În plus, multe alte pachete software (proiectate pentru stațiile de lucru UNIX cu sistemul X Window) pot fi portate ușor și sub Linux, grație compatibilității sistemului Linux cu standarde de portabilitate cum sunt POSIX.1 sau sistemul X Window.



Observație

O altă dezvoltare interesantă o constituie suportul pentru standardul iBCS (Intel Binary Compatibility Standard) sub Linux. Standardul iBCS oferă compatibilitate binară între aplicațiile dezvoltate pe diferite platforme UNIX care au la bază procesoare Intel x86. Spre exemplu, suportul pentru iBCS va permite rularea pe sistemul dumneavoastră Linux a variantei SCO UNIX a programului WordPerfect.

Suportul pentru iBCS poate fi activat sub forma unui modul (un modul este similar unui program rezident din DOS), care poate fi încărcat după pornirea sistemului Linux utilizând o comandă de forma:

```
/sbin/insmod -f /lib/modules/2.0.31/misc/iBCS
```

Distribuția Red Hat Linux de pe CD-ROM-ul atașat include modulul iBCS și documentația corespunzătoare acestuia sub forma unui fișier RPM. Puteți instala acest fișier de pe CD-ROM-ul Red Hat Linux parcurgând următoarele etape:

1. Introduceți CD-ul cu sistemul Red Hat Linux în unitatea CD-ROM și montați unitatea:

```
mount /dev/hdc /cdrom
```

Amintiți-vă să înlocuiți */dev/hdc* cu numele de dispozitiv corespunzător unității CD-ROM. Dacă directorul */cdrom* nu există, îl puteți crea cu comanda `mkdir /cdrom`.

2. Deplasați-vă în directorul */cdrom/RedHat/RPMS* și utilizați apoi comanda `rpm -U` pentru instalarea modulului iBCS:

```
cd /cdrom/RedHat/RPMS
rpm -U iBCS*
```

După cum se poate afla și din fișierul README din directorul */usr/doc/iBCS-2.0*, modulul iBCS permite încărcarea și rularea fișierelor executabile din următoarele sisteme de operare:

- SCO UNIX (inclusiv SCO OpenServer 5)
- Solaris (versiunea x86)
- System V Release 4 pentru procesoare Intel (variante printre care se numără Unix Ware, Dell UNIX sau Interactive UNIX)
- Wyse V/386 UNIX
- Xenix V/386
- Xenix 286

În prezent se lucrează la asigurarea suportului pentru alte sisteme de operare, printre care 386BSD, FreeBSD, NetBSD și BSDI/386.

Sistemul X Window sub Linux

Trebuie să admitem că ne plictisim ușor atunci când ne limităm la introducerea unor comenzi de la tastatură. Cei care cunosc deja pe dinăfară un număr mare de comenzi s-ar putea să nu realizeze, dar sistemul UNIX nu se va răspândi foarte mult decât dacă va deveni ușor de utilizat. În acest punct, sistemul X Window, sau X, ne vine în ajutor.

Sistemul X oferă un mecanism standard de afișare a imaginilor „hartă de biți” independente de dispozitiv. X include de asemenea și un sistem de ferestre, permițând aplicațiilor să își afișeze rezultatele în ferestre separate.

Chiar dacă X pune la dispoziție mecanismul de afișare în ferestre separate, aplicațiile suferă prin lipsa unui aspect specific, distinctiv. Acest aspect specific este dat de interfețele grafice (Graphical User Interface – GUI) cum ar fi OSF/Motif și OPEN LOOK, care utilizează sistemul X Window. Dintre acestea două, Motif a acaparat segmentul majoritar de piață, în timp ce interfața OPEN LOOK continuă să fie disponibilă pe stațiile firmei Sun Microsystems.



Referință încrucișată

Distribuția Red Hat Linux de pe CD-ROM-ul atașat este livrată împreună cu sistemul X Window în varianta XFree86 3.3.1 – o implementare a X11R6 (sistemul X Window, versiunea 11, varianta 6 – cea mai recentă a sistemului X) pentru arhitecturi 80x86. O facilitate esențială a sistemului XFree86 este suportul pentru un număr mare de plăci video disponibile în prezent pentru calculatoarele PC. După cum veți vedea în capitolul 4, XFree86 suportă câteva sute de plăci video, de la adaptoarele SVGA (Super Video Graphics Adapter) și până la plăcile accelerate cum sunt cele alcătuite pe baza cipseturilor video S3, Mach64 sau Weitek P9000.

În ceea ce privește interfața grafică, sistemul Linux include mai multe *programe de gestiune a ferestrelor* (window managers), care permit celorlalte programe X să își afișeze rezultatele în ferestre separate. Programele de gestiune a ferestrelor adaugă de obicei marginea și bara de titlu la fereastra unei aplicații X.

Chiar dacă Motif domină piața interfețelor grafice UNIX, această interfață nu intră în componența sistemului Linux, deoarece firma Open Software Foundation nu o distribuie gratuit. Interfața Motif are un aspect similar cu cel oferit de Microsoft Windows și conține atât un program special de gestiune a ferestrelor – Motif Window Manager (mwm), cât și setul de instrumente Motif pentru dezvoltarea de programe. Puteți obține versiunea pentru Linux a interfeței Motif de la mai multe firme distribuitoare de programe.

Deși obținerea interfeței Motif presupune un cost suplimentar, dacă aveți nevoie de un mediu grafic de dezvoltare a proiectelor software, un calculator Linux cu interfața Motif reprezintă una dintre cele mai ieftine soluții. Dacă aveți o firmă de consultanță sau dacă doriți să dezvoltați programe X sau Motif acasă, sistemul Linux este cu siguranță cea mai bună alegere.

Spre deosebire de Microsoft Windows sau sistemul Macintosh, Linux nu oferă un utilitar grafic pentru gestiunea fișierelor, deci este foarte probabil că va fi necesar să învățați anumite comenzi UNIX și să utilizați o fereastră terminal sub X pentru a lucra cu fișiere. Puteți însă utiliza sistemul Linux și XFree86 pentru a dezvolta programe grafice ușor de utilizat chiar și de către începători.

În distribuția sistemului Linux, sunt incluse câteva aplicații grafice care rulează sub X. Cele mai importante astfel de programe sunt cele legate de afișarea și editarea imaginilor. Primul este utilitarul XV, realizat de John Bradley și distribuit ca shareware; un altul este xpaint, realizat de David Koblas; un al treilea este GIMP, un program cu facilități similare cu cele oferite de Adobe Photoshop.

Un alt aspect important legat de sistemul X Window este acela că permite rularea aplicațiilor în rețea. De exemplu, este posibil să rulați o aplicație grafică pe un alt server din rețea, vizualizând însă rezultatele acesteia și interacționând cu aplicația respectivă de pe propria suprafață de lucru. Cu alte cuvinte, sistemul X permite PC-ului dumneavoastră să devină o poartă către toate celelalte sisteme din rețea.



Referință încrucișată

Capitolul 4 conține mai multe detalii despre modul de configurare a sistemului XFree86, iar capitolul 23 prezintă modul în care puteți dezvolta aplicații X pe sistemul dumneavoastră.

Lucrul în rețea sub Linux

Termenul de *lucru în rețea* se referă la toate aspectele legate de schimbul de date între două sau mai multe calculatoare, pornind de la conexiunea fizică și până la protocolul care permite realizarea efectivă a schimbului de date. Un *protocol de rețea* reprezintă o metodă prin care un transmițător și un receptor convin să realizeze un schimb de date în cadrul unei rețele.

În cadrul diferitelor niveluri ale unei rețele se utilizează protocoale de rețea distincte. La nivelul fizic – nivelul la care biții de date traversează un mediu fizic cum ar fi un cablu – se utilizează protocoale cum sunt Ethernet sau token-ring. Programele de aplicații nu lucrează în general direct la nivelul fizic. Acestea se bazează pe protocoale care știu să opereze cu blocuri de date. Printre aceste protocoale se numără și Internet Packet Exchange (IPX) oferit de firma Novell, cât și binecunoscutul Transmission Control Protocol/Internet Protocol (TCP/IP).



Referință încrucișată

Diferitele niveluri ale protocoalelor de rețea pot fi reprezentate printr-un model al rețelei. Unul dintre acestea este modelul de referință cu șapte niveluri Open Systems Interconnection (OSI) dezvoltat de Organizația Internațională pentru Standardizare (International Standards Organization – ISO). Acest model este analizat în capitolul 16.

Protocoalele standard de rețea, cum este TCP/IP, stau la baza creșterii amănunțite a numărului de calculatoare interconectate, rezultatul fiind apariția atât a unor rețele locale, cât și a unor extinse pe arii largi (Wide Area Networks – WAN). Aceste protocoale au permis interconectarea acestor rețele de dimensiuni mai mici, rezultând rețele de dimensiuni mari care formează o interrețea: Internet-ul.

TCP/IP

Lucrul în rețea a fost punctul forte al sistemelor UNIX încă de la apariția acestora. În particular, familia de protocoale TCP/IP a făcut parte din sistemele UNIX încă de la apariția în varianta BSD UNIX în 1982. În momentul actual, TCP/IP reprezintă protocolul WAN utilizat în întregul Internet.

Sistemul Linux oferă suport pentru familia de protocoale TCP/IP și include de asemenea cele mai cunoscute aplicații de lucru în rețea, cum sunt telnet, ftp sau rlogin. La nivel fizic, sistemul Linux include driver-e pentru un număr mare de plăci de rețea. Protocolul token-ring este de asemenea implementat în sursele nucleului sistemului Linux. Pentru a utiliza acest protocol, este suficient să reconstruiți nucleul, activând suportul pentru token-ring.



Referință încrucișată

Se poate afirma că suportul pentru TCP/IP – familia de protocoale cel mai răspândită în Internet – reprezintă o facilitate naturală a sistemului Linux. Dezvoltarea rapidă a sistemului Linux însuși nu ar fi fost posibilă fără colaborarea unui număr mare de programatori localizați în mare parte în Europa și America de Nord. Această colaborare a fost însă posibilă numai datorită existenței Internet-ului. În

capitolul 19 veți afla cum puteți configura o rețea TCP/IP și cum puteți utiliza principalele programe de lucru în rețea.

Sistemul Linux include de asemenea și interfața Berkeley Sockets (numele provine de la faptul că aceasta a apărut inițial în versiunea Berkeley UNIX în 1982) – o modalitate foarte răspândită de programare a aplicațiilor în rețelele TCP/IP. Pentru cei care programează în limbajul C, această interfață constă din câteva fișiere antet și un număr de funcții C utile pentru stabilirea conexiunilor și transmitia și recepția datelor.

Puteți utiliza interfața Berkeley Sockets pentru a dezvolta instrumente de lucru în Internet, cum sunt browserele World Wide Web. Cum majoritatea programelor TCP/IP (inclusiv cele disponibile în mod gratuit pe diferitele situri din Internet) utilizează interfața Sockets, configurarea și utilizarea acestor programe sub Linux este de regulă ușoară, deoarece sistemul Linux include interfața Sockets.

PPP și SLIP

Nu oricine dispune de o conexiune Ethernet la rețeaua Internet – mai ales cei care utilizează Linux pe calculatoarele personale de acasă. Există însă o modalitate de conectare la Internet și de comunicare utilizând protocoalele TCP/IP pe o linie telefonică la care este conectat un modem. Aveți nevoie de un *server* – un sistem conectat la Internet – care acceptă conexiuni de la sistemul dumneavoastră.

În prezent, firme comerciale denumite *furnizori de servicii Internet* (Internet Service Providers – ISP), oferă acest serviciu contra unei taxe. Dacă nu doriți să plătiți o taxă pentru conexiunea la Internet, vă puteți interesa dacă un calculator aflat la firma unde lucrați oferă acest acces. Această opțiune nu este nerezonabilă, mai ales atunci când lucrați acasă în domeniul dezvoltării de programe UNIX.

Serverul care permite accesul la Internet rulează unul din protocoalele:

- Serial Line Internet Protocol (SLIP)
- Point-to-Point Protocol (PPP)

Ambele protocoale suportă TCP/IP pe o linie comutată. Protocolul SLIP este mai simplu și mai vechi decât PPP, care dispune însă de mai multe opțiuni pentru stabilirea unei conexiuni. În prezent, aproape toată lumea utilizează PPP. Pentru a stabili o conexiune, sistemul dumneavoastră trebuie să ruleze același protocol ca și sistemul furnizorului de servicii.



Referință încrucișată

Sistemul Linux suportă atât SLIP, cât și PPP pentru realizarea unor conexiuni la Internet prin linia comutată. Puteți de asemenea să configurați propriul sistem Linux ca server SLIP sau PPP, astfel încât alte calculatoare să îl poată apela pe al dumneavoastră pentru a stabili o conexiune TCP/IP prin linia telefonică. Capitolul 18 prezintă modul în care puteți configura SLIP și PPP pe sistemul dumneavoastră Linux.

Partajarea fișierelor și NFS

Din lumea MS-DOS și Microsoft Windows se cunoaște conceptul de *server de fișiere* – un sistem care memorează fișiere importante și permite celorlalte sisteme din rețea să aibă acces la acele fișiere. În esență, toate PC-urile din rețea partajează unul sau mai multe discuri centrale. În DOS sau Windows, utilizatorii văd discul serverului de fișiere ca pe o altă unitate, identificată printr-o literă distinctă (de exemplu U). În rețelele de PC-uri, partajarea fișierelor este de regulă implementată prin protocoalele Novell NetWare și Microsoft LAN Manager.

Conceptul de partajare a fișierelor există de asemenea și sub UNIX. *Sistemul de fișiere în rețea* (Network File System – NFS) oferă un mecanism standard prin care un sistem poate avea acces la fișierele altui sistem din rețea. Pentru utilizator, fișierele sistemului de la distanță apar ca și cum ar fi într-un director din sistemul local.



Referință încrucișată

Sistemul NFS este disponibil și sub Linux. Puteți partaja directoarele sistemului dumneavoastră cu alte sisteme care suportă NFS. Celelalte sisteme care au acces la fișierele sistemului dumneavoastră nu trebuie neapărat să ruleze UNIX. De fapt, sistemul NFS este disponibil și sub DOS sau Windows; prin urmare, puteți utiliza PC-ul dumneavoastră ca server de fișiere pentru un grup de PC-uri care rulează DOS și Windows. Capitolul 21 prezintă detaliat posibilitatea de utilizare a unui sistem Linux ca server de fișiere.

UUCP

Un protocol destul de vechi, dar important, care permite schimbul de date este UUCP (UNIX-to-UNIX Copy). Pentru anumite sisteme, acest protocol continuă să fie un mijloc prin care transferă poșta electronică sau știrile. Știrile Usenet – și totodată sistemele de dirijare a știrilor (Bulletin Board Systems – BBS) din Internet – își au originea în UUCP. Calculatoarele care erau conectate prin linii telefonice și modemi utilizau UUCP pentru transferul poștei electronice, știrilor sau chiar fișierelor. În esență, mesajele și știrile erau dirijate de la un calculator la celălalt. Acest sistem reprezenta o soluție ieftină pentru transferul mesajelor și știrilor. Chiar dacă în prezent poșta electronică și știrile parcurg în majoritatea cazurilor conexiuni permanente la Internet, UUCP permite încă multor sisteme aflate la distanță să fie conectate la comunitatea Internet-ului prin intermediul știrilor USENET și poștei electronice.

Sistemul Linux include protocolul UUCP. Dacă sistemul dumneavoastră dispune de un modem și doriți să transferați fișiere aflate pe un alt sistem prin intermediul unei conexiuni pe linie comutată, puteți utiliza UUCP. Datorită însă proliferării firmelor furnizoare de servicii Internet, este foarte probabil ca sistemul dumneavoastră să fie conectat la Internet prin intermediul unei astfel de firme.

Administrarea sistemului Linux

Prin *administrare a sistemului* ne referim în general la activitățile care trebuie efectuate pentru a menține un calculator în stare de funcționare corectă. Cum în prezent majoritatea sistemelor sunt conectate la rețea, un alt set de activități este necesar pentru menținerea conexiunii la rețea în stare de funcționare. Aceste activități sunt denumite global prin sintagma de *administrare a rețelei*. Un sit cu mai multe calculatoare dispune în general de un *administrator de sistem*, persoană angajată în scopul de se ocupa de activitățile de administrare a sistemelor și rețelei. Siturile mari dispun chiar de personal separat pentru administrarea sistemelor și rețelei. Dacă rulați însă Linux numai pe un singur PC sau pe câteva sisteme dintr-o firmă mai mică, probabil că vă veți ocupa atât de administrarea sistemelor, cât și a rețelei.



Referință încrucișată

Sistemul Linux conține toate comenzile și utilitarele de bază necesare pentru administrarea sistemului și rețelei. Capitolul 6 prezintă pe scurt câteva dintre aceste comenzi. Instrumentele de administrare a rețelei sunt prezentate în capitolele 18 și 19.

Activități de administrare a unui sistem

Ca administrator de sistem, activitățile dumneavoastră obișnuite trebuie să cuprindă:

■ **Instalarea, configurarea și actualizarea sistemului de operare și diferitelor utilitare.** Ați învățat să instalați sistemul Linux și alte pachete software în capitolul 1. Capitolul 2 prezintă modul în care puteți actualiza sistemul Linux, iar capitolul 4 descrie modul de instalare pentru sistemul X Window.

■ **Adăugarea și ștergerea utilizatorilor.** După cum s-a arătat în secțiunea „Adăugarea conturilor utilizatorilor” din capitolul 1, puteți utiliza instrumentul grafic User Configurator pentru a adăuga noi utilizatori după ce instalați sistemul Linux. Dacă un anumit utilizator își uită parola, trebuie să utilizați comanda `passwd` pentru a-i modifica parola.

■ **Instalarea noilor programe.** Pentru programele Linux obișnuite, pe care le obțineți sub formă de cod sursă, această activitate necesită utilizarea unor instrumente cum sunt `gunzip` (pentru decompimarea programelor), `tar` (pentru despachetarea arhivelor) și `make` (pentru a construi programele executabile). În cazul programelor distribuite de firma Red Hat sub formă de fișiere Red Hat Package Manager (RPM), trebuie să utilizați comanda `rpm` pentru a efectua instalarea. Utilitarul RPM este prezentat în capitolul 2.

■ **Activitățile de salvare.** Puteți utiliza programul `tar` pentru a arhiva unul sau mai multe directoare și pentru a copia arhiva pe o dischetă (dacă aceasta încapă pe o dischetă) sau pe o bandă (dacă dispuneți de unitate de bandă).

■ **Montarea și demontarea sistemelor de fișiere.** Atunci când doriți să citiți o dischetă MS-DOS, spre exemplu, trebuie să montați sistemul de fișiere de pe dischetă într-unul din directoarele sistemului de fișiere Linux. Pentru a realiza aceasta, trebuie să utilizați comanda `mount`.

■ **Monitorizarea performanțelor funcționării sistemului.** În acest scop, trebuie să utilizați câteva programe, cum ar fi `top` (pentru a determina activitățile care consumă majoritatea timpului procesorului) și `free` (pentru a determina cantitatea de memorie disponibilă și utilizată din sistem).

■ **Pornirea și oprirea sistemului.** Chiar dacă pornirea sistemului nu presupune de regulă decât alimentarea acestuia, trebuie să aveți grijă atunci când opriți sistemul. Trebuie să utilizați comanda `shutdown` pentru a opri toate programele în rulare înainte de a întrerupe alimentarea PC-ului.

Activitățile administratorului de rețea

Activitățile obișnuite de administrare a rețelei cuprind:

■ **Adăugarea suportului pentru rețea în nucleu.** Trebuie să configurați și să reconstruiți nucleul Linux astfel încât să includă suport pentru diferitele plăci de rețea, ca și pentru PPP sau SLIP. Capitolul 2 vă prezintă etapele pe care trebuie să le parcurgeți la configurarea și construirea nucleului.

■ **Gestiunea fișierelor de configurare a rețelei.** Sub Linux (ca și în cazul altor sisteme UNIX), rețeaua TCP/IP este configurată prin intermediul câtorva fișiere text pe care trebuie să le editați pentru ca rețeaua să poată funcționa. Este necesar să editați unul sau mai multe din următoarele fișiere: `/etc/hosts`, `/etc/networks`, `/etc/host.conf`, `/etc/resolv.conf`, `/etc/HOSTNAME`, `/etc/hosts.allow`, `/etc/hosts.deny` și script-urile din directorul `/etc/sysconfig/network-scripts`.

■ **Configurarea protocoalelor PPP și SLIP.** Pentru configurarea conexiunilor PPP și SLIP, veți utiliza instrumente cum sunt `chat` sau `dip`.

■ **Monitorizarea stării rețelei.** Trebuie să utilizați comenzi cum sunt `netstat` (pentru a vizualiza informații referitoare la conexiunile active în rețea), `/sbin/ifconfig` (pentru a verifica starea diferitelor interfețe de rețea) și `ping` (pentru a vă asigura că o anumită conexiune funcționează).

■ **Configurarea protocolului UUCP** (dacă doriți să contactați prin linie comutată un alt sistem pentru transferul poștei electronice sau știrilor). Este foarte probabil că nu veți utiliza deloc UUCP, din cauză că majoritatea sistemelor transferă în prezent poșta electronică sau știrile prin intermediul protocolului TCP/IP direct din Internet.

DOS și Linux

După cum deja știți, cele mai răspândite sisteme de operare pentru calculatoarele PC 80386, 80486 sau Pentium sunt MS-DOS și Microsoft Windows. Din cauză că la început sistemul Linux a fost destinat procesoarelor 80386/80486, o legătură între Linux și DOS a existat întotdeauna. De regulă, instalarea sistemului Linux începe cu aceiași pași ca și în DOS.

Conexiunea dintre Linux și DOS s-a menținut în mai multe moduri:

■ Linux suportă sistemul de fișiere MS-DOS. Din Linux, puteți avea acces la fișierele MS-DOS de pe hard-disc sau de pe o dischetă.

■ Sistemul Linux cuprinde un set de instrumente (denumit `mtools`) care permite lucrul cu fișiere MS-DOS din Linux.

■ Se află în lucru un emulator MS-DOS (denumit `DOSEMU`) care utilizează modul virtual 8086 al procesoarelor 80386 sau ulterioare la fel cum sistemul Windows 95 oferă posibilitatea rulării unei sesiuni DOS într-o anumită fereastră.

Proiectul DOSEMU a înregistrat deja un anumit succes, multe aplicații DOS (printre care WordPerfect 5.1 și FoxPro 2.0) putând fi rulate sub emulatorul DOS.

Un alt proiect, denumit WINE, încearcă să dezvolte un emulator Windows sub sistemul X Window din Linux. De fapt, WINE va permite utilizarea unei instalări curente a sistemului Windows 3.1 și rularea programelor Windows 3.1 instalate pe partiția DOS a PC-ului dumneavoastră.



Referință
încrucșată

Capitolul 7 prezintă modul în care puteți avea acces la fișierele DOS de sub Linux. Capitolul prezintă de asemenea utilitarele `mtools` și oferă informații despre DOSEMU.

Dezvoltarea programelor în Linux

Din toate posibilitățile de utilizare, dezvoltarea de programe este probabil cea mai potrivită. Instrumentele de dezvoltare de programe, cum ar fi compilatoarele și bibliotecile, sunt disponibile, acestea fiind oricum necesare pentru reconstruirea nucleului sistemului. Dacă vă ocupați cu dezvoltarea programelor sub UNIX, cunoașteți deja sistemul UNIX, deci în Linux vă veți simți ca acasă.

În ceea ce privește dezvoltarea de programe, dispuneți de aceleași instrumente de bază (cum ar fi un editor, un compilator și un depanator) ca și pe alte stații de lucru UNIX, cum ar fi cele ale firmelor Hewlett-Packard (HP), Sun Microsystems, IBM sau Digital Equipment Corporation (DEC). Prin urmare, dacă în cursul zilei lucrați pe o astfel de stație UNIX, puteți utiliza acasă un PC pe care ați instalat Linux, care oferă același mediu de dezvoltare cu un cost mult redus. Puteți astfel să realizați proiecte complexe acasă, sau vă puteți dedica timpul liber programelor pe care le scrieți de plăcere și pe care apoi le distribuiți gratuit pe Internet.

Pentru a ne forma o imagine despre suportul oferit de Linux pentru dezvoltarea de programe, prezentăm în continuare o listă a diverselor facilități care fac ca Linux să fie un mediu productiv pentru dezvoltarea de programe:

- Compilatorul de C GNU, gcc, care poate compila programe ANSI-C.
- Compilatorul de C++ GNU (g++), care oferă suport pentru facilitățile AT&T C++ 3.0.
- Depanatorul GNU, gdb, care permite execuția pas cu pas a unui program pentru determinarea problemelor apărute și locurilor unde programul eșuează. (Imaginea în memorie a unui program care a eșuat este salvată într-un fișier numit core; gdb poate examina acest fișier.)
- Utilitarul gprof, care permite determinarea gradului în care un anumit program utilizează timpul procesorului.
- Utilitarele GNU make și imake, care permit compilarea și legarea programelor mari.
- Un sistem de control al versiunilor (Revision Control System – RCS), care memorează informațiile referitoare la versiuni și controlează accesul la fișierele sursă, astfel încât doi programatori să nu poată modifica inadvertent un anumit fișier sursă.
- Editorul GNU Emacs, care pregătește fișierele sursă și chiar lansează procesul de compilare pentru a construi un program.
- Limbajul de comenzi Perl, utilizat pentru a scrie programe care leagă mai multe programe de dimensiuni reduse cu comenzile UNIX, în scopul efectuării unei anumite activități.
- Limbajul Tool Command Language și setul de instrumente pentru X al acestuia (Tcl/Tk), care permit proiectarea rapidă a aplicațiilor X.
- Biblioteci dinamice partajate, care permit micșorarea considerabilă a dimensiunilor programelor executabile, deoarece toate bibliotecile necesare mai multor programe sunt partajate, o copie a acestora fiind încărcată în memoria sistemului.
- Fișiere antet și biblioteci POSIX.1 și POSIX.2, care permit dezvoltarea de programe portabile.



Referință încrucișată

Capitolul 22 prezintă dezvoltarea de programe în Linux. Parcurgeți mai întâi capitolul 8, în care veți învăța despre programarea în Tcl/Tk.

Sistemul Linux – primul pas spre Internet

Mulți dintre dumneavoastră aveți deja acces la Internet și ați experimentat resursele pe care această rețea le oferă:

Poșta electronică, grupurile de discuții și rețeaua World Wide Web (WWW). Indiferent dacă motto-ul dumneavoastră a fost „Veni vidi vici” sau „Ce este WWW?”, veți fi încântat să aflați că sistemul dumneavoastră conține tot ce îi trebuie pentru accesul la Internet. De fapt, PC-ul dumneavoastră poate deveni un membru cu drepturi depline al societății Internet-ului, cu propriul server Web, pe care puteți publica orice informații doriți.

Chiar dacă sistemul Linux include TCP/IP și toate celelalte programe de rețea cu ajutorul cărora vă puteți configura PC-ul ca host Internet, există totuși o problemă: mai întâi trebuie să obțineți o conexiune fizică la Internet. Sistemul dumneavoastră trebuie să fie conectat la un alt *nod* (care poate fi un alt calculator sau un echipament de rețea, de exemplu un router) din Internet. Aceasta este marea problemă pentru majoritatea celor care dețin sisteme Linux – o conexiune la Internet este de regulă costisitoare, prețul fiind proporțional cu viteza de transfer a informației.

Mulți furnizori comerciali de Internet oferă diferite tipuri de conexiuni fizice la Internet. În România, cu prețuri cuprinse între echivalentul a 9 și 30 \$ S.U.A. puteți obține un cont pe un server PPP. Puteți apoi rula programul PPP și pe sistemul dvs: mai întâi apăsați serverul, cu ajutorul unui modem; vă puteți conecta apoi la Internet la viteze cuprinse între 28800 biți pe secundă (bps) și 56000 bps, în funcție de modemul de care dispuneți.

Deși o conexiune prin linie comutată este suficientă ca să aveți acces la Internet, să primiți poșta electronică sau să citiți știrile, în mod cert nu mai este satisfăcătoare atunci când doriți ca sistemul să ofere informații și pentru alți utilizatori prin intermediul rețelei Web sau al protocolului FTP (File Transfer Protocol). Pentru aceasta, aveți nevoie de o conexiune disponibilă 24 de ore pe zi, din cauza faptului că alte sisteme pot încerca să îl contacteze pe al dumneavoastră în permanență. Pentru o taxă echivalentă cu câteva sute de dolari S.U.A. pe lună, puteți obține o legătură SLIP sau PPP dedicată, transformând sistemul dumneavoastră într-o prezență permanentă pe Internet.

O altă posibilitate de afacere – pentru cineva care dispune de câteva PC-uri conectate în rețea – este conectarea unei întregi rețele locale (LAN) la Internet. Puteți rula sistemul Linux pe unul dintre PC-uri pentru a stabili conexiunea. De regulă, dispuneți de un LAN Ethernet rulând TCP/IP, la care sunt conectate toate mașinile din rețea, inclusiv sistemul Linux. Sistemul Linux stabilește o conexiune SLIP sau PPP la Internet (prin intermediul unei linii comutate sau închiriate). Puteți configura apoi sistemul Linux ca poartă între LAN-ul Ethernet și Internet, astfel încât PC-urile din LAN să aibă acces la celelalte sisteme din Internet.



Referință încrucișată

În capitolul 19 veți învăța cum puteți configura sistemul Linux pentru a avea acces la Internet. Mai mult, veți afla și cum puteți utiliza un sistem Linux ca poartă între rețeaua locală și Internet.

Rezumat

După ce ați instalat și ați configurat sistemul Linux pe PC-ul dumneavoastră, vă puteți orienta atenția către activitățile pe care v-ați gândit să le rezolvați cu ajutorul lui. Indiferent că doriți să dezvoltați programe sau să vă transformați PC-ul într-un sit Internet, puteți utiliza sistemul Linux cu încredere dacă îi stăpâniți facilitățile generale. În consecință, acest capitol face o trecere în revistă a diferitelor aspecte legate de Linux, începând cu dezvoltarea de programe și până la lucrul în rețea și administrarea de sistem. Următorul capitol revine la o altă activitate de configurare, prezentând modul în care puteți configura și rula sistemul X Window pe calculatorul dumneavoastră. Configurarea sistemului X vă pune la dispoziție o interfață grafică sub Linux.

Din acest capitol ați reținut următoarele:

- Comunitatea celor care dezvoltă sistemul Linux utilizează un algoritm de numerotare a versiunilor, pentru a vă ajuta să înțelegeți semnificația diferitelor versiuni. Versiunile 2.x.y, în care x este număr par, reprezintă versiuni stabile. Numărul y reprezintă *nivelul de patch*, care este incrementat pe măsură ce anumite probleme sunt rezolvate. Versiunile 2.x.y cu un număr x impar reprezintă versiuni beta destinate celor care asigură dezvoltarea sistemului; acestea pot avea o funcționare instabilă.
- POSIX este abrevierea de la Portable Operating System Interface (completată astfel încât să rimeze cu UNIX). Institutul Inginerilor Electrotehniști și Electroniști (IEEE) a

promovat dezvoltarea standardelor POSIX pentru a asigura portabilitatea aplicațiilor între diferitele medii UNIX.

- ▶ Linux reprezintă un sistem de operare din familia UNIX, conformându-se standardului POSIX.1. Prin urmare, Linux reprezintă sistemul UNIX ideal din punct de vedere al costului.
- ▶ Distribuția Linux de pe CD-ROM-ul atașat include de asemenea un modul numit Intel Binary Compatibility Standard (iBCS), pe care îl puteți instala cu ajutorul comenzii `rpm` și apoi încărca utilizând comanda `/sbin/insmod`. Cu ajutorul acestui modul, sistemul Linux poate rula programele binare din alte sisteme UNIX destinate procesoarelor din familia Intel x86, cum sunt UnixWare, SCO Unix, Dell UNIX, Interactive UNIX, Xenix V/386 și Xenix 286.
- ▶ Distribuția Linux atașată lucrării de față include pachetul software XFree86 (sistemul X Window versiunea 11, varianta 6, sau X11R6). După ce veți instala XFree86, veți dispune de o interfață grafică pentru sistemul Linux. În plus, sistemul X vă permite să rulați aplicații în rețea – adică să rulați aplicații pe un alt sistem din rețea, afișând rezultatele acestora pe monitorul PC-ului dumneavoastră.
- ▶ Sistemul Linux oferă suport pentru protocoalele TCP/IP. TCP/IP reprezintă protocolul de rețea utilizat la nivelul întregului Internet. Un PC cu sistemul Linux este prin urmare ideal ca sit Internet, oferind diverse servicii Internet, cum ar fi FTP sau World Wide Web. Puteți utiliza PC-ul dumneavoastră ca poartă spre Internet, conectându-vă la un furnizor de servicii printr-o linie comutată și rulând un browser Web pentru a naviga pe rețea.
- ▶ Distribuția Linux conține toate instrumentele pentru dezvoltarea de programe necesare pentru a scrie aplicații UNIX sau X. Dispuneți de compilatoarele GNU pentru C și C++ pentru a compila fișiere sursă, de `make` pentru automatizarea activității de compilare, de depanatorul `gdb` pentru detectarea erorilor și de sistemul RCS pentru gestiunea diferitelor versiuni ale unui fișier. Prin urmare, un calculator Linux reprezintă o stație de lucru ideală pentru oricine se arată interesat în dezvoltarea de aplicații.

Capitolul 4

Secretele sistemului X sub Linux

În acest capitol:

- ▶ O privire generală asupra sistemului X
- ▶ Instalarea XFree86 sub Linux
- ▶ Configurarea XFree86
- ▶ Pornirea sistemului X
- ▶ Experimentarea diferitelor moduri video
- ▶ Părăsirea sistemului X

Dacă ați utilizat unul din sistemele de operare Macintosh System sau Microsoft Windows, cunoașteți desigur câteva aspecte legate de interfețele grafice. În universul stațiilor de lucru UNIX, interfața grafică nu reprezintă o parte integrantă a sistemului de operare. În locul acesteia, stațiile UNIX utilizează de regulă Motif ca interfață grafică. Interfața *Motif*, pe de altă parte, este construită pe baza unui sistem de ferestre numit *X Window*, sau pe scurt *X*. Sistemul Linux oferă de asemenea suport pentru X, incluzând o versiune de X numită XFree86, proiectată astfel încât să poată lucra cu multitudinea de plăci video destinate calculatoarelor PC. În ceea ce privește interfața grafică, Linux conține mai multe programe de gestiune a ferestrelor.



Referință încrucișată

Acest capitol prezintă sistemul X Window și oferă detaliile necesare pentru instalarea XFree86 pe un sistem Linux. CD-ROM-ul atașat conține pachetul XFree86, pe care l-ați instalat deja pe discul dumneavoastră în decursul procesului descris în capitolul 1. Tot atunci ați și configurat XFree86. Acest capitol își propune să prezinte în detaliu fișierul de configurare pentru XFree86 și totodată descrie ce trebuie să faceți pentru a lansa X pe PC-ul dumneavoastră.

Ce reprezintă sistemul X Window

Termenul de *X Window* descrie în mare un număr de componente care permit afișarea în ferestre grafice pe o mulțime de sisteme de afișare de tip „hartă de biți”.

Esența sistemului X este *serverul X* – un proces (program) care rulează pe un calculator echipat cu un sistem de afișare de tip „hartă de biți”, o tastatură și un mouse. Aplicațiile – *clienții X* – care efectuează anumite afișări, inițiază un schimb de informații cu serverul X prin intermediul unuia dintre mecanismele cunoscute de comunicare între procese. Comunicația dintre clienții X și serverul X se efectuează conform regulilor unui protocol bine determinat, numit *protocolul X*. În afară de serverul X și de protocolul X, termenul de X se mai referă și la o bibliotecă de funcții cunoscută sub numele de *Xlib*, care constituie interfața în limbajul C cu facilitățile oferite de serverul X.

Sisteme de afișare de tip „hartă de biți”

Sistemele de afișare de tip „hartă de biți” sunt compuse din două părți distincte:

- Un monitor video – de regulă, bazat pe un tub catodic, care permite afișarea în mod grafic. Monitorul este de regulă denumit *ecran de afișare* sau pur și simplu *ecran*.

- O placă video (sau placă grafică) – aceasta este ori o placă separată, ori o mulțime de circuite încorporate în placa de bază a sistemului, care determină apariția imaginii prin trimiterea semnalelor corespunzătoare către monitor.

Într-un sistem de afișare de tip „hartă de biți”, monitorul afișează un tablou de puncte (denumite *pixeli*, sau *elemente de imagine*), aspectul fiecărui pixel fiind în corespondență cu conținutul unei locații de memorie din memoria plăcii video (numită pe scurt memorie video). Pentru un monitor alb-negru, în care fiecare pixel poate fi alb sau negru, starea unui pixel poate fi reprezentată cu ajutorul unui singur bit. Termenul de „hartă de biți” se referă la corespondența dintre fiecare bit din memoria video și un pixel de pe ecran.

Pentru monitoarele color, fiecare pixel poate fi reprezentat în memorie printr-un număr de 4 până la 24 de biți. Numărul de biți pe pixel este denumit *adâncime* a unui pixel. Adâncimea determină numărul de culori care pot fi afișate simultan pe ecran. Pentru o adâncime de 8 biți, spre exemplu, se pot afișa simultan $2^8 = 256$ de culori.

Plăcile video pot opera în diferite moduri video, fiecare dintre acestea având o altă valoare a adâncimii. Majoritatea plăcilor video suportă o adâncime

de 8 biți, unele fiind capabile să suporte și adâncimi de 24 de biți.

Imaginile „hartă de biți” sunt denumite și *imagini rastru*, din cauză că imaginile care apar pe monitor sunt construite dintr-un număr mare de linii orizontale, denumite *linii de rastru*. Aceste linii sunt generate de către un fascicul de electroni care balează un ecran acoperit cu un strat de fosfor. Cum fiecare grăunte de fosfor, corespunzător unui pixel, strălucește proporțional cu intensitatea fasciculului, fiecare linie a imaginii poate fi generată controlând intensitatea fasciculului pe măsură ce acesta balează ecranul. Un monitor creează iluzia unei imagini statice prin redesenarea repetată a liniilor de rastru. Majoritatea monitoarelor redesează întregul ecran de linii de rastru de 50 până la 90 de ori în fiecare secundă.

Sistemele de afișare color reprezintă orice culoare utilizând o combinație a trei culori primare: roșu (R), verde (G) și albastru (B). Termenul de valoare RGB este deseori utilizat pentru a specifica o culoare prin intensitățile culorilor primare. Un monitor color utilizează trei fascicule distincte de electroni – câte unul pentru fiecare culoare primară. Ecranul unui monitor color dispune de un model triunghiular repetat de puncte fosforescente având culorile primare: roșu, verde și albastru. Fiecare punct fosforescent strălucește în culoarea proprie atunci când fasciculul corespunzător de electroni se ciocnește de el. Placa video trimite semnale care controlează intensitatea fasciculelor de electroni, determinând astfel afișarea pe monitor a multor nuanțe coloristice.

Clienți și servere

Atunci când lucrați cu un sistem de afișare X, serverul X rulează pe calculatorul dumneavoastră și controlează monitorul, tastatura și mouse-ul. Serverul răspunde comenzilor trimise de clienții X care deschid ferestre pe ecran și desenează în acele ferestre. Acest model este cunoscut sub numele de model *client/server*. După cum o spune și numele, serverul oferă un anumit serviciu necesar clienților. De regulă, comunicația între clienți și server are loc în rețea, schimbul de date urmând un protocol înțeles de ambele entități.

Probabil că ați văzut deja modelul client/server la lucru. Un *server de fișiere*, spre exemplu, stochează fișiere și permite clienților să aibă acces și să opereze asupra acelor fișiere. O altă aplicație uzuală o reprezintă un *server de baze de date*, care oferă o bază de date centralizată

din care clienții pot extrage datele necesare formulând anumite interogări. Similar, serverul X oferă servicii de afișare clienților care îi trimit cereri ce se conformează protocolului X.

Există însă o diferență majoră între serverul X și celelalte servere, cum ar fi serverele de fișiere sau de baze de date. În timp ce serverele de fișiere sau de baze de date reprezintă procese care se execută pe mașini aflate la distanță, serverul X este un proces care se execută pe mașina la care este atașat monitorul. Clienții X se pot executa atât local, cât și pe sisteme aflate la distanță.



Observație

În modelul client/server al unei rețele locale de PC-uri, aplicațiile sunt de regulă păstrate de un server central. Utilizatorii au acces la aceste aplicații ale serverului de pe PC-urile care reprezintă clienții. Atunci când rulați o aplicație de pe un PC, acea aplicație se execută pe procesorul PC-ului respectiv. În modelul client/server al sistemului X, situația este diferită. Atunci când rulați o aplicație, aceasta este rulată de către procesorul serverului – numai rezultatele rulării sunt afișate pe sistemul dumneavoastră (care rulează serverul X). Cu alte cuvinte, serverul X rulează într-o locație pe care de regulă o asociem clienților (în situația PC-urilor dintr-o rețea care reprezintă clienți). Este important să vă amintiți acest amănunt atunci când lucrați cu modelul client/server în X.

Interfețele grafice și sistemul X

Interfața grafică a unei aplicații determină atât aspectul acesteia, cât și comportamentul. Atunci când interfața utilizează obiecte grafice, cum ar fi ferestre sau meniuri, este denumită *interfață grafică cu utilizatorii* (mulți o numesc pe scurt GUI). Acest tip de interfață este numit „point-and-click”, deoarece utilizatorii interacționează în general deplasând cursorul mouse-ului pe ecran și executând clic pe butonul mouse-ului. Pentru verificarea închiderii unui fișier de exemplu, utilizatorul va executa clic cu butonul mouse-ului în timp ce cursorul acestuia se găsește în interiorul unei casete etichetate cu OK.

Interfețele grafice au fost dezvoltate pentru prima dată la centrul de cercetare Xerox din Palo Alto (PARC). Ulterior, firma Apple Computer a dezvoltat astfel de interfețe pentru sistemele de operare Lisa și Macintosh. În zilele noastre, interfețele grafice sunt disponibile pe majoritatea sistemelor. Microsoft Windows deține supremația în ceea ce privește calculatoarele compatibile IBM PC cu procesoare 386 sau mai bune; în sistemul OS/2, avem interfața Presentation Manager; în fine, în sistemele UNIX dispunem de interfețele OSF/Motif și OPEN LOOK (construite pe infrastructura sistemului X).

Majoritatea interfețelor grafice, inclusiv Motif și OPEN LOOK, au patru componente de bază:

- Un sistem de ferestre
- Un program de gestiune a ferestrelor
- Un set de instrumente de dezvoltare
- Un stil specific

Sistemul de ferestre grafic organizează afișarea imaginilor pe ecran și execută operațiile de bază de desenare a textului și imaginilor.

Programul de gestiune a ferestrelor permite utilizatorului să deplaseze ferestre și să schimbe dimensiunile acestora. Tot acest program răspunde și de aspectul ferestrelor, deoarece le adaugă un chenar decorativ specific. Un alt aspect important rezolvat de acest program este gestiunea *selecției intrării*: acesta este mecanismul prin care utilizatorul poate selecta una dintre ferestrele afișate pe ecran, care devine *fereastră activă* (fereastra cu care utilizatorul va interacționa). Acest proces este denumit selecție a unei ferestre. Pentru interfețele grafice

construite peste sistemul X Window, programul de gestiune a ferestrelor este un client X, la fel ca și oricare altă aplicație X.

A treia componentă, setul de instrumente, reprezintă o bibliotecă de funcții cu o interfață de programare bine stabilită. Acest set de instrumente este în primul rând de interes pentru programatori, deoarece le permite acestora să scrie aplicații care utilizează facilitățile sistemului de ferestre și au totodată un comportament compatibil cu aspectul.

La prima vedere, ai putea crede că aceste trei componente – sistemul de ferestre, programul de gestiune și setul de instrumente – sunt suficiente pentru a alcătui o interfață. Lucrurile nu stau însă tocmai așa. Dacă programatorii nu urmează un set comun de reguli, aspectul și comportamentul aplicațiilor construite cu ajutorul unei interfețe grafice nu vor fi compatibile. Prin urmare, o interfață grafică necesită o a patra componentă esențială: *stilul specific*, care definește aspectul și comportamentul interfeței unei aplicații.

Un scurt istoric al sistemului X

Dezvoltarea sistemului X Window a început în 1984 la Institutul pentru Tehnologie din Massachusetts (MIT), sub conducerea Laboratorului de Informatică, în cadrul proiectului Athena. La început, proiectul X a beneficiat de sprijinul unor firme ca DEC sau IBM, implicate în proiectul Athena. La începutul lui 1986, firma DEC a lansat pe piață prima versiune comercială a sistemului X pe stațiile VAX-station-11/GPX sub sistemul de operare Ultrix: versiunea 10, varianta 3 a sistemului X (X10R3). În scurt timp, sistemul X a atras atenția și altor firme producătoare de stații de lucru, printre care Hewlett-Packard, Apollo Computer (firma Apollo a fuzionat între timp cu Hewlett-Packard), Sun Microsystems și Tektronix.

Cerințele formulate de utilizatorii sistemului X10 au determinat membrii proiectului să înceapă lucrul la o nouă versiune majoră a protocolului X. În decembrie 1986, simultan cu demararea proiectului care a devenit ulterior versiunea 11 a sistemului X (X11), era lansată varianta X10R4, ultima a sistemului X10.

În ianuarie 1987, la prima conferință tehnică dedicată proiectului X, 11 firme importante fabricante de calculatoare și-au anunțat unirea eforturilor pentru realizarea unei standardizări a sistemului X11. Prima variantă a sistemului X11 – X11R1 – a fost

lansată în septembrie 1987. Pentru a se asigura continuitatea evoluției sistemului X sub controlul unei organizații deschise, în ianuarie 1988 s-a format la MIT consorțiul X. Sub conducerea lui Robert W. Scheifler, unul dintre principalii arhitecți ai sistemului X, consorțiul a reprezentat un factor important în asigurarea succesului proiectului X. De atunci, controlul asupra dezvoltării sistemului X Window a fost preluat de firma X Consortium Inc., o organizație non-profit.

În martie 1988 a fost lansată varianta 2 a sistemului X11 – X11R2. Varianta 3 – X11R3 – a apărut la finele lunii decembrie 1988. Consorțiul X a lansat X11R4 în ianuarie 1990; varianta X11R5 l-a urmat în august 1991.

Pe măsură ce sistemul X a prins rădăcini în lumea stațiilor de lucru, consorțiul X a continuat să îmbunătățească sistemul din mai multe puncte de vedere, printre care suportul pentru programarea sub X în limbajul C++ și adăugarea unui set de instrumente orientate pe obiect numit Fresco, care a făcut parte din varianta X11R6 (lansată în aprilie 1994). În toate aceste variante, protocolul X a rămas nemodificat. Toate îmbunătățirile au fost făcute pornind de la capabilitatea protocolului de a permite extensii. În momentul apariției lucrării, varianta cea mai utilizată a sistemului X era X11R6.

Se poate crede că această cerință de înscriere într-un stil specific stă în calea creativității programatorilor; în realitate, această cerință se referă numai la elementele comune ale interfeței unei aplicații. Stilul stabilește convențiile de bază, cum ar fi locațiile meniurilor File, Edit și Help în bara de meniuri sau semnificația legării butoanelor de mouse (prin legare înțelegem

aici asocierea dintre apăsarea unui buton și o anumită acțiune, cum ar fi apariția unui meniu atunci când utilizatorul apasă butonul drept). Interfața grafică nu impune însă nici un fel de restricții asupra funcțiilor specifice pe care aplicația le implementează. Programatorii au astfel șanse nelimitate de a-și afirma creativitatea în proiectarea părților specifice ale aplicațiilor.

Sistemul X sub Linux

Sistemul X Window sub Linux este rezultatul proiectului XFree86 – un proiect colectiv al multor programatori, având scopul de a implementa sistemul X pentru PC-uri. În consecință, versiunea pentru Linux a sistemului X se numește XFree86.

Serverul X răspunde de afișarea imaginii pe monitor. Prin urmare, serverul trebuie să poată utiliza placa video. În lumea calculatoarelor PC, se utilizează o gamă largă de plăci video; crearea unui server X care să poată lucra cu toate aceste plăci este dificilă. Cipseturile (circuiturile integrate) comune sunt însă utilizate de multe tipuri de plăci video. Proiectul XFree86 pune la dispoziție mai multe servere, fiecare dintre acestea fiind capabile să lucreze cu un anumit cipset.

Din nefericire, chiar dacă mai multe firme pot construi plăci video bazate pe un același cipset, aceste firme pot configura cipsetul în moduri distincte. În consecință, serverele XFree86 trebuie să fie parametrizate pentru a putea lucra cu fiecare marcă specifică de plăci video. La început, câteva firme producătoare de plăci video au refuzat să ofere informații despre produsele lor. Masa în continuă creștere a utilizatorilor Linux a exercitat asupra acestor firme o presiune continuă în scopul de a obține informațiile necesare pentru proiectul XFree86; ca atare, versiunea XFree86 3.3.1 suportă un număr mare de plăci video care nu au fost disponibile în versiunile anterioare.



Observație

Versiunea sistemului XFree86 de pe CD-ROM-ul atașat suportă seriile de plăci Diamond SpeedStar și Diamond Stealth. Acest suport a devenit disponibil în momentul în care firma Diamond a fost de acord să pună la dispoziția celor care dezvoltă sistemul XFree86 informațiile tehnice necesare.

Configurarea sistemului X sub Linux

Desigur că doriți să configurați și să utilizați cât mai repede sistemul X – fără acesta, sistemul Linux nu dispune de o interfață grafică. Dacă sunteți deja obișnuit cu alte medii grafice (de pe alte tipuri de stații UNIX sau de pe un PC rulând Microsoft Windows), doriți cu siguranță să utilizați un mediu grafic similar și sub Linux.

Dacă aveți intenția de a dezvolta programe sub Linux, este foarte posibil ca programele pe care doriți să le realizați să aibă o parte cu caracter grafic, care trebuie testată și implementată sub X. Pentru aceasta, trebuie mai întâi să configurați și să rulați sistemul X.

Indiferent de ceea ce vă propuneți, atunci când doriți să configurați XFree86, trebuie să pregătiți un fișier de configurare special, numit `XFree86Config`, care conține informații referitoare la configurația hardware a sistemului dumneavoastră. Începând cu versiunea 3.1.2, sistemul XFree86 dispune de un program utilitar numit `x86config` (numele este identic cu al fișierului de configurare, dar este scris cu litere mici), care vă poate ajuta să creați fișierul `XFree86Config`.

Următoarele secțiuni constituie un ghid în procesul de configurare a sistemului XFree86 și lansare a serverului X pe sistemul dumneavoastră.

Ce trebuie să știți despre configurația sistemului dumneavoastră înainte de a configura XFree86

Pentru a configura sistemul XFree86, trebuie să cunoașteți dispozitivele fizice pe care sistemul X trebuie să le utilizeze. Din scurta introducere a acestui capitol în X, știți că serverul X controlează următoarele dispozitive:

- Placa video
- Monitorul
- Tastatura
- Mouse-ul

Serverul X are nevoie de informații despre fiecare dintre aceste componente pentru a funcționa corespunzător.

Monitorul

Sistemul XFree86 controlează monitorul prin intermediul plăcii video. Ca atare, un server XFree86 poate determina o placă video să trimită un spectru larg de semnale către monitor (pentru a controla de exemplu viteza de afișare a unei linii de rastru sau viteza de redesenare a întregului ecran). Dacă o placă video determină monitorul să efectueze o activitate care îi depășește posibilitățile (cum ar fi desenarea unei linii de rastru la o viteză mult mai mare decât aceea pentru care monitorul a fost proiectat), acesta se poate strica. Pentru a vă asigura că semnalele trimise de placa video sunt în domeniul acceptabil pentru monitor, sistemul XFree86 necesită informații referitoare la anumite caracteristici importante ale monitorului.

Informația minimală pe care trebuie să o cunoașteți despre monitorul dumneavoastră conține următorii parametri:

- Spectrul acceptabil de valori pentru frecvența de sincronizare pe orizontală. Valorile obișnuite sunt cuprinse între 30 și 64 kilohertzi (kHz).
- Domeniul acceptabil pentru ratele de sincronizare pe verticală (cunoscute și sub numele de *rate de reîmprospătare pe verticală*), cuprins de regulă între 50 și 90 de hertzi (Hz).
- Dacă este disponibilă, lărgimea de bandă, exprimată în megahertzi, spre exemplu 75 MHz.

De regulă, documentația monitorului include toate aceste informații. Dacă ați cumpărat de curând calculatorul, aveți încă documentația. Dacă ați pierdut documentația monitorului, un mod de a afla informații este din configurația sistemului Microsoft Windows. Dacă sistemul a fost livrat împreună cu un driver Windows pentru sistemul de afișare, acel driver va afișa anumite informații despre monitor. Utilitarul Norton System Information poate de asemenea să ofere această informație. O altă soluție este de a vizita situl Web al firmei de la care ați cumpărat calculatorul, căutând specificațiile tehnice ale monitorului. Există cazuri în care această metodă a fost încununată de succes.

Placa video

Sistemul XFree86 pune la dispoziție servere X care cunosc un anumit cipset video (circuitul video care generează semnalele necesare pentru a controla monitorul). Pentru a selecta în mod corect serverul X, trebuie să indicați exact cipsetul pe care placa video îl utilizează.

Chiar și în cazul unei familii de plăci video bazate pe un același cipset, există mulți parametri care variază de la o placă la alta. Prin urmare, trebuie de asemenea să precizați numele firmei producătoare și modelul plăcii video.

Informația minimală pe care trebuie să o cunoașteți despre placa video se rezumă deci la:

- cipsetul video, cum ar fi S3 sau ATI Mach64
- numele firmei producătoare și modelul, cum ar fi Diamond Stealth 64 VRAM, Number Nine GXE64 sau ATI Graphics Xpression
- Cantitatea de memorie RAM (memorie cu acces aleator), cum ar fi 1 MB sau 2 MB.

Majoritatea firmelor producătoare indică numai marca și modelul unei plăci video în reclame; acestea menționează numai rareori și cipsetul video. Trebuie să cereți în mod explicit informațiile despre cipsetul video, cât și orice alte informații pe care firma producătoare este dispusă să le ofere despre modelul plăcii.

Dacă rulați Linux pe un PC mai vechi, puteți încerca să găsiți aceste informații deschizând carcasa calculatorului și examinând placa video. Numele firmei producătoare și numărul modelului sunt de regulă inscripționate pe placă. Cât despre cipsetul video, trebuie să examinați inscripțiile de pe diferitele circuite de pe placă, încercând să ghiciți care dintre ele reprezintă numele cipsetului. Pe un anumit model de placă video, este posibil să găsiți un circuit cu următoarele inscripții (reproduse doar parțial în cele ce urmează):

S3 Trio 64 (GACC 2)
86C764 - P

Puteți deduce că această placă utilizează cipsetul S3. În realitate, inscripțiile de pe circuit arată și numărul 86C764 (în formă completă sau prescurtată, ca 764). Dacă ați dedus numele firmei producătoare și modelul plăcii, veți putea configura și rula serverul X pe PC-ul dumneavoastră.

Mouse-ul

Mouse-ul reprezintă o parte integrantă a oricărei interfețe grafice, datorită faptului că utilizatorii își pot exprima opțiunile sau pot executa anumite activități prin deplasarea și apăsarea cursorului său. Serverul X realizează deplasarea unui cursor pe ecran sincron cu deplasarea mouse-ului de către utilizator. De asemenea, serverul X detectează toate apăsările butoanelor mouse-ului și trimite evenimentele generate de acestea aplicațiilor client cărora acestea le sunt destinate – adică aplicațiilor în ale căror ferestre se găsește cursorul mouse-ului.

Chiar dacă ați configurat deja mouse-ul la instalarea sistemului Linux, trebuie să oferiți informațiile despre mouse și serverul X. Serverul XFree86 are neapărat nevoie de mouse pentru a putea porni; dacă serverul X nu poate avea acces la un mouse, nu se va lansa în execuție.

Pentru a descrie un mouse, trebuie să cunoașteți următoarele informații:

- Tipul mouse-ului, care poate fi Microsoft, Logitech, BusMouse sau PS/2
- Tipul de conexiune cu sistemul dumneavoastră – serială sau direct la magistrala sistemului
- Numele dispozitivului asociat mouse-ului. Puteți accepta numele implicit `/dev/mouse`, deoarece programul de instalare al sistemului Red Hat leagă acest nume simbolic de numele adevărat al dispozitivului asociat. Numele real depinde de tipul mouse-ului și locul în care acesta este conectat – pentru un mouse serial, de exemplu, numărul portului serial.

Dacă ați specificat corect tipul mouse-ului și numele dispozitivului asociat, acesta trebuie să funcționeze corect.



Atenție!

Dacă aveți un mouse de magistrală, se recomandă să știți că rularea programului `gpm` (care vă permite să lucrați cu mouse-ul în modul text) poate interfera cu serverul X. În acest caz, este bine să vă asigurați că la pornirea serverului X, programul `gpm` nu

se află în rulare. Pentru a detecta dacă `gpm` rulează sau nu, utilizați următoarea comandă și căutați procesul `gpm` în rezultatul afișat:

```
ps ax| grep gpm
```

Dacă programul `gpm` rulează, utilizați comanda `gpm -k` pentru a-l opri înainte de pornirea serverului X.

Utilizarea programului `xf86config`

Serverele X XFree86 citesc și interpretează fișierul `XF86Config` pentru a afla informații detaliate despre anumite părți componente ale sistemului dumneavoastră: monitorul, placa video, tastatura și mouse-ul. În plus, fișierul `XF86Config` precizează modurile video pe care doriți să le utilizați.

Generarea fișierului `XF86Config` se face de regulă utilizând un program numit `xf86config`. Acest program vă pune un set de întrebări la care trebuie să răspundeți. Pentru aceasta, trebuie să cunoașteți câte ceva despre placa video și monitorul sistemului dumneavoastră.

Pentru a rula programul `xf86config`, intrați în sistem ca root și tastați `xf86config` la promptul sistemului Linux. Programul `xf86config` va afișa un ecran în mod text care conține următoarele informații:

- Fișierul `XF86Config` se găsește în directorul `/etc/X11`, sau în `/usr/X11R6/lib/X11`. În directorul `/usr/X11R6/lib/X11` se găsește o configurație model, salvată în fișierul numit `XF86Config.eg`; această configurație presupune existența unei plăci video standard VGA și a unui monitor cu rezoluția de 640 x 480.
- Puteți edita fișierul `XF86Config` sau puteți crea un nou fișier, executând în continuare programul `xf86config`.
- Fișierul `/usr/X11R6/lib/X11/doc/README.config` prezintă operația de configurare.
- Fișierele `README` (cu rol informativ) pentru cipseturile video specifice se găsesc în directorul `/usr/X11R6/lib/X11/doc`.
- Înainte de a lansa programul `xf86config`, este necesar să cunoașteți cipsetul video și dimensiunea memoriei video.

Citind secțiunea precedentă, ați aflat aproape tot ce se putea despre placa video și monitorul din configurația dumneavoastră. Apăsați tasta Enter pentru a continua.

Programul `xf86config` vă cere să vă asigurați că directorul `/usr/X11R6/bin` există în variabila de mediu `PATH` (valoarea acestei variabile determină ordinea în care sistemul caută anumite fișiere binare pentru a le executa). Dacă ați instalat sistemul XFree86 de pe CD-ROM-ul atașat, directorul `/usr/X11` reprezintă o legătură simbolică – un alt nume – pentru directorul `/usr/X11R6`, ceea ce înseamnă că vă puteți referi la directorul `/usr/X11/bin` pentru a desemna de fapt directorul `/usr/X11R6/bin`.

Programul `xf86config` afișează de asemenea valoarea curentă a variabilei de mediu `PATH`. Cum această valoare conține directorul `/usr/X11R6/bin`, puteți apăsa tasta Enter pentru a trece la pasul următor.

Programul `xf86config` vă cere să precizați un protocol pentru mouse, parametru care determină modul în care se va desfășura comunicația dintre serverul X și mouse. Programul vă prezintă următoarele opțiuni:

1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol)
3. Bus Mouse

4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
9. Microsoft IntelliMouse

Apăsați tasta care corespunde tipului mouse-ului dumneavoastră. Dacă mouse-ul este conectat la un port PS/2, alegeți cifra 4. Apăsați tasta corespunzătoare și mai departe Enter.

Programul `xf86config` vă pune apoi următoarea întrebare:

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons?

Dacă mouse-ul are două butoane, programul `xf86config` recomandă să activați opțiunea de simulare a celui de-al treilea. Apăsați `y` și apoi Enter. Dacă activați această opțiune, puteți simula funcționarea butonului din mijloc prin apăsarea simultană a celor două butoane. Multe aplicații X presupun că mouse-ul are trei butoane, astfel încât această facilitare este extrem de utilă în lumea calculatoarelor PC, în care un mouse are de regulă numai două butoane.

În continuare, trebuie să precizați numele complet al dispozitivului atașat mouse-ului. La instalarea sistemului Linux, atunci când ați configurat mouse-ul, programul de instalare a creat o legătură între numele real al dispozitivului și numele standard `/dev/mouse`; puteți apăsa deci tasta Enter. Pentru un mouse serial, utilizați unul din următoarele nume:

- `/dev/ttyS0` pentru un mouse conectat la portul COM1
- `/dev/ttyS1` pentru un mouse conectat la portul COM2
- `/dev/ttyS2` pentru un mouse conectat la portul COM3
- `/dev/ttyS3` pentru un mouse conectat la portul COM4

În cazul unui mouse de magistrală, se puteți alege dintre următoarele nume:

- `/dev/atibm` pentru un mouse ATI
- `/dev/logibm` pentru un mouse Logitech
- `/dev/inportbm` pentru un mouse Microsoft
- `/dev/psaux` pentru orice mouse conectat la portul auxiliar PS/2.

Introduceți numele dispozitivului corespunzător mouse-ului și apăsați apoi Enter. Este permis să introduceți numele exact, chiar dacă există o legătură între `/dev/mouse` și numele real al dispozitivului.

Programul `xf86config` vă întreabă în continuare dacă doriți să utilizați extensia `XKEYBOARD` (o facilitare a sistemului X11R6) pentru gestiunea planului tastaturii (planul tastaturii determină asocierea dintre caractere și tastele existente):

Please answer the following question with either 'y' or 'n'.
Do you want to use XKB? n

Dacă răspundeți cu `Yes`, trebuie să selectați unul dintre tipurile predefinite de planuri pentru tastatură. Se recomandă să răspundeți cu `n`; puteți utiliza apoi programul `xmodmap` pentru a altera planul tastaturii.

Programul `xf86config` vă întreabă apoi dacă doriți să generați caracterele speciale atunci când lucrați în X. Pentru aceasta, puteți lega tasta Alt din stânga cu Meta, iar tasta Alt din dreapta cu ModeShift. (*Meta* și *ModeShift* reprezintă numele standardizate ale tastelor speciale utilizate în X; puteți asocia aceste taste speciale cu orice tastă fizică.) Întrebarea este formulată astfel:

Please answer the following question with either 'y' or 'n'.
Do you want to enable these bindings for the Alt keys? y

Activarea acestor taste nu prezintă nici un dezavantaj, deci apăsați y și apoi Enter pentru a continua configurarea.

În continuare, xf86config vă informează că are nevoie de doi parametri esențiali ai monitorului dumneavoastră:

■ Frecvența de sincronizare pe orizontală – ori de câte ori poate afișa monitorul o linie raster orizontală în fiecare secundă, exprimat în kilohertzi (kHz)

■ Rata de sincronizare pe verticală – de câte ori pe secundă monitorul poate reafixa întregul ecran, valoare exprimată în hertzi (Hz)

Aceste informații se regăsesc în manualul monitorului.

Apăsați tasta Enter pentru a continua. Programul xf86config vă oferă următoarele opțiuni:

```
hsync in kHz; monitor type with characteristic modes
1 31.5; Standard VGA, 640x480 @ 60 Hz
2 31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
3 31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
4 31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
5 31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
6 31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
7 31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
8 31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
9 31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10 31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11 Enter your own horizontal sync range
```

Puteți introduce numărul corespunzător monitorului dumneavoastră, sau puteți alege 11 pentru a specifica un domeniu.



Atenție!

Nu specificați un domeniu de sincronizare pe orizontală care să depășească posibilitățile monitorului. O valoare eronată poate conduce la defectarea monitorului.

Majoritatea manualelor monitoarelor precizează un domeniu de valori pentru rata de sincronizare pe orizontală. Pentru a introduce un domeniu propriu, tastați 11 și apoi Enter. Programul vă cere în acest caz un domeniu de valori. Introduceți acest domeniu sub forma a două valori, separate prin semnul minus (-). Dacă manualul monitorului dumneavoastră precizează un domeniu de sincronizare pe orizontală cuprins între 30-64 kHz, trebuie să introduceți:

Horizontal sync range: 30-64

În continuare, xf86config vă cere să alegeți rata de sincronizare pe verticală dintr-o listă care conține următoarele opțiuni:

```
1 50-70
2 50-90
3 50-100
4 40-150
5 Enter your own vertical sync range
```

Dacă știți cu exactitate domeniul, apăsați 5 și apoi Enter. La următorul prompt, introduceți domeniul pentru rata de sincronizare pe verticală. De exemplu, dacă documentația monitorului indică un domeniu cuprins între 55 și 90 Hz, veți introduce:

Vertical sync range: 55-90

În cele ce urmează, trebuie să introduceți un identificator pentru definiția monitorului dumneavoastră. De regulă, se recomandă să introduceți marca și modelul monitorului. Puteți însă introduce orice valori aici, aceste informații fiind utilizate numai cu scopul de a identifica monitorul într-o altă secțiune a fișierului de configurare XF86Config (structura acestui fișier va fi prezentată în secțiunea următoare). O posibilitate de răspuns este:

```
Enter an identifier for your monitor definition: Dell VS15X
Enter the vendor name of your monitor: Dell
Enter the model name of your monitor: VS15X
```

Următoarea activitate o constituie configurarea plăcii video. Programul xf86config afișează un mesaj explicativ și apoi pune următoarea întrebare:

Do you want to look at the card database ? y

Apăsați y și apoi Enter. Programul va afișa o listă de 381 de plăci. Apăsați Enter după fiecare ecran pentru a vizualiza întreaga listă. Lista completă a plăcilor este prezentată în continuare. Marca, modelul și cipsetul unei plăci video reprezintă informații esențiale de care aveți nevoie pentru a selecta în mod corect serverul XFree86 și pentru a configura corect acest server. Parcurgerea listei este necesară pentru a vă informa dacă sistemul X va putea utiliza placa dumneavoastră.

0 2 the Max MAXColor S3 Trio64V+	S3 Trio64V+
1 928Movie	S3 928
2 AGX (generic)	AGX-014/15/16
3 ALG-5434 (E)	CL-GD5434
4 ASUS PCI-AV264CT	ATI-Mach64
5 ASUS PCI-V264CT	ATI-Mach64
6 ASUS Video Magic PCI V864	S3 864
7 ASUS Video Magic PCI VT64	S3 Trio64
8 ATI 3D Pro Turbo	ATI-Mach64
9 ATI 3D Xpression	ATI-Mach64
10 ATI 3D Xpression+ PC2TV	ATI-Mach64
11 ATI 8514 Ultra (no VGA)	ATI-Mach8
12 ATI All-in-Wonder	ATI-Mach64
13 ATI Graphics Pro Turbo	ATI-Mach64
14 ATI Graphics Pro Turbo 1600	ATI-Mach64
15 ATI Graphics Ultra	ATI-Mach8
16 ATI Graphics Ultra Pro	ATI-Mach32
17 ATI Graphics Xpression with 68875 RAMDAC	ATI-Mach64

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

18 ATI Graphics Xpression with AT&T 20C408 RAMDAC	ATI-Mach64
19 ATI Graphics Xpression with CH8398 RAMDAC	ATI-Mach64
20 ATI Graphics Xpression with Mach64 CT (264CT)	ATI-Mach64
21 ATI Graphics Xpression with STG1702 RAMDAC	ATI-Mach64
22 ATI Mach64	ATI-Mach64
23 ATI Mach64 3D RAGE II+, Internal RAMDAC	ATI-Mach64
24 ATI Mach64 3D RAGE II, Internal RAMDAC	ATI-Mach64
25 ATI Mach64 CT (264CT), Internal RAMDAC	ATI-Mach64
26 ATI Mach64 GT (264GT), aka 3D RAGE, Internal RAMDAC	ATI-Mach64
27 ATI Mach64 VT (264VT), Internal RAMDAC	ATI-Mach64
28 ATI Mach64 with AT&T 20C408 RAMDAC	ATI-Mach64
29 ATI Mach64 with CH8398 RAMDAC	ATI-Mach64
30 ATI Mach64 with IBM RGB514 RAMDAC	ATI-Mach64
31 ATI Ultra Plus	ATI-Mach32
32 ATI Video Xpression	ATI-Mach64
33 ATI Win Boost with AT&T 20C408 RAMDAC	ATI-Mach64

34 ATI Win Boost with CH8398 RAMDAC	ATI-Mach64
35 ATI Win Boost with Mach64 CT (264CT)	ATI-Mach64

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

36 ATI Win Boost with STG1702 RAMDAC	ATI-Mach64
37 ATI Win Turbo	ATI-Mach64
38 ATI Wonder SVGA	ATI vgaWonder
39 ATrend ATC-2165A	ET6000
40 Actix GE32+ 2MB	S3 801/805
41 Actix GE32i	S3 805i
42 Actix GE64	S3 864
43 Actix ProStar	CL-GD5426/5428
44 Actix ProStar 64	CL-GD5434
45 Actix Ultra	S3 928
46 Acumos AVGA3	CL-GD5420/2/4/6/8/9
47 Alliance ProMotion 6422	AP6422
48 Ark Logic ARK1000PV (generic)	ARK1000PV
49 Ark Logic ARK1000VL (generic)	ARK1000VL
50 Ark Logic ARK2000MT (generic)	ARK1000MT
51 Ark Logic ARK2000PV (generic)	ARK1000PV
52 Avance Logic 2101	Avance Logic
53 Avance Logic 2228	Avance Logic

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

54 Avance Logic 2301	Avance Logic
55 Avance Logic 2302	Avance Logic
56 Avance Logic 2308	Avance Logic
57 Avance Logic 2401	Avance Logic
58 Binar Graphics AnyView	ET6000
59 Boca Vortex (Sierra RAMDAC)	AGX-015
60 California Graphics SunTracer 6000	ET6000
61 Canopus Co. Power Window 3DV	S3 ViRGE
62 Canopus Total-3D	Verite 1000
63 Cardex Challenger (Pro)	ET4000/W32(i/p)
64 Cardex Cobra	ET4000/W32(i/p)
65 Cardex Trio64	S3 Trio64
66 Cardex Trio64Pro	S3 Trio64
67 Chips & Technologies CT64200	ct64200
68 Chips & Technologies CT64300	ct64300
69 Chips & Technologies CT65520	ct65520
70 Chips & Technologies CT65525	ct65525
71 Chips & Technologies CT65530	ct65530

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

72 Chips & Technologies CT65535	ct65535
73 Chips & Technologies CT65540	ct65540
74 Chips & Technologies CT65545	ct65545
75 Chips & Technologies CT65546	ct65546
76 Chips & Technologies CT65548	ct65548
77 Chips & Technologies CT65550	ct65550
78 Chips & Technologies CT65554	ct65554
79 Chips & Technologies CT65555	ct65555
80 Chips & Technologies CT68554	ct68554
81 Cirrus Logic GD542x	CL-GD5420/2/4/6/8/9
82 Cirrus Logic GD543x	CL-GD5430/5434

83 Cirrus Logic GD544x	CL-GD544x
84 Cirrus Logic GD5462	CL-GD5462
85 Cirrus Logic GD5464	CL-GD5464
86 Cirrus Logic GD62xx (laptop)	CL-GD6205/15/25/35
87 Cirrus Logic GD64xx (laptop)	CL-GD6420/6440
88 Cirrus Logic GD754x (laptop)	CL-GD7541/42/43/48
89 Colorgraphic Dual Lightning	ET4000/W32(i/p)

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

90 Creative Labs 3D Blaster PCI (Verite 1000)	Verite 1000
91 Creative Labs Graphics Blaster 3D	CL-GD5464
92 Creative Labs Graphics Blaster MA201	CL-GD544x
93 Creative Labs Graphics Blaster MA202	CL-GD544x
94 Creative Labs Graphics Blaster MA302	CL-GD5462
95 Creative Labs Graphics Blaster MA334	CL-GD5464
96 DFI-WG1000	CL-GD5420/2/4/6/8/9
97 DFI-WG5000	ET4000/W32(i/p)
98 DFI-WG6000	WD90C33
99 DSV3325	S3 ViRGE
100 DSV3326	S3 Trio64V+
101 DataExpert DSV3325	S3 ViRGE
102 DataExpert DSV3365	S3 Trio64V+
103 Dell S3 805	S3 801/805
104 Dell onboard ET4000	ET4000
105 Diamond Edge 3D	nv1
106 Diamond Multimedia Stealth 3D 2000	S3 ViRGE
107 Diamond Multimedia Stealth 3D 2000 PRO	S3 ViRGE/DX

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

108 Diamond SpeedStar (Plus)	ET4000
109 Diamond SpeedStar 24	ET4000
110 Diamond SpeedStar 24X (not fully supported)	WD90C31
111 Diamond SpeedStar 64	CL-GD5434
112 Diamond SpeedStar HiColor	ET4000
113 Diamond SpeedStar Pro (not SE)	CL-GD5426/28
114 Diamond SpeedStar Pro 1100	CL-GD5420/2/4/6/8/9
115 Diamond SpeedStar Pro SE (CL-GD5430/5434)	CL-GD5430/5434
116 Diamond SpeedStar64 Graphics 2000/2200	CL-GD5434
117 Diamond Stealth 24	S3 801/805
118 Diamond Stealth 32	ET4000/W32(i/p)
119 Diamond Stealth 3D 2000	S3 ViRGE
120 Diamond Stealth 3D 2000 PRO	S3 ViRGE/DX
121 Diamond Stealth 3D 3000	S3 ViRGE/VX
122 Diamond Stealth 64 DRAM SE	S3 Trio32
123 Diamond Stealth 64 DRAM with S3 SDAC	S3 864
124 Diamond Stealth 64 DRAM with S3 Trio64	S3 Trio64
125 Diamond Stealth 64 VRAM	S3 964

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

126 Diamond Stealth 64 Video VRAM (TI RAMDAC)	S3 968
127 Diamond Stealth Pro	S3 928
128 Diamond Stealth VRAM	S3 911/924
129 Diamond Stealth Video 2500	Alliance AT24
130 Diamond Stealth Video DRAM	S3 868
131 Diamond Stealth64 Graphics 2001 series	ARK2000PV

132 Diamond Stealth64 Graphics 2xx0 series (864 + SDAC) S3 864
 133 Diamond Stealth64 Graphics 2xx0 series (Trio64) S3 Trio64
 134 Diamond Stealth64 Video 2001 series (2121/2201) S3 Trio64V+
 135 Diamond Stealth64 Video 2120/2200 S3 868
 136 Diamond Stealth64 Video 3200 S3 968
 137 Diamond Stealth64 Video 3240/3400 (IBM RAMDAC) S3 968
 138 Diamond Stealth64 Video 3240/3400 (TI RAMDAC) S3 968
 139 Diamond Viper PCI 2mb Weitek 9000
 140 Diamond Viper VLB 2mb Weitek 9000
 141 EIZO (VRAM) AGX-014/15/16
 142 ELSA Gloria-4 S3 968
 143 ELSA Gloria-8 S3 968

Enter a number to choose the corresponding card definition.
 Press Enter for the next page; press q to continue configuration.

144 ELSA Victory 3D S3 ViRGE
 145 ELSA Victory 3DX S3 ViRGE/DX
 146 ELSA WINNER 1000/T2D S3 Trio64V2
 147 ELSA Winner 1000AVI (AT&T 20C409 version) S3 868
 148 ELSA Winner 1000AVI (SDAC version) S3 868
 149 ELSA Winner 1000ISA S3 805i
 150 ELSA Winner 1000PRO with S3 SDAC S3 864
 151 ELSA Winner 1000PRO with STG1700 or AT&T RAMDAC S3 864
 152 ELSA Winner 1000PRO/X S3 868
 153 ELSA Winner 1000TRIO S3 Trio64
 154 ELSA Winner 1000TRIO/V S3 Trio64V+
 155 ELSA Winner 1000TwinBus S3 928
 156 ELSA Winner 1000VL S3 928
 157 ELSA Winner 2000 S3 928
 158 ELSA Winner 2000AVI S3 968
 159 ELSA Winner 2000AVI/3D S3 ViRGE/VX
 160 ELSA Winner 2000PRO-2 S3 964
 161 ELSA Winner 2000PRO-4 S3 964

Enter a number to choose the corresponding card definition.
 Press Enter for the next page; press q to continue configuration.

162 ELSA Winner 2000PRO/X-2 S3 968
 163 ELSA Winner 2000PRO/X-4 S3 968
 164 ELSA Winner 2000PRO/X-8 S3 968
 165 ELSA Winner 3000 S3 ViRGE/VX
 166 ELSA Winner 3000-L-42 S3 ViRGE/VX
 167 ELSA Winner 3000-M-22 S3 ViRGE/VX
 168 ELSA Winner 3000-S S3 ViRGE
 169 ET3000 (generic) ET3000
 170 ET4000 (generic) ET4000
 171 ET4000 W32i, W32p (generic) ET4000/W32(i/p)
 172 ET4000/W32 (generic) ET4000/W32
 173 ET6000 (generic) ET6000
 174 ExpertColor DSV3325 S3 ViRGE
 175 ExpertColor DSV3365 S3 Trio64V+
 176 Generic VGA compatible Generic VGA
 177 Genoa 5400 ET3000
 178 Genoa 8500VL(-28) CL-GD5426/28
 179 Genoa 8900 Phantom 32i ET4000/W32(i/p)

Enter a number to choose the corresponding card definition.
 Press Enter for the next page; press q to continue configuration.

180 Genoa Phantom 64i with S3 SDAC S3 864

181 Genoa VideoBlitz III AV S3 968
 182 Hercules Dynamite ET4000/W32
 183 Hercules Dynamite 128/Video ET6000
 184 Hercules Dynamite Power ET4000/W32(i/p)
 185 Hercules Dynamite Pro ET4000/W32(i/p)
 186 Hercules Graphite HG210 AGX-014
 187 Hercules Graphite Power AGX-016
 188 Hercules Graphite Pro AGX-015
 189 Hercules Graphite Terminator 64 S3 964
 190 Hercules Graphite Terminator 64/DRAM S3 Trio64
 191 Hercules Graphite Terminator Pro 64 S3 968
 192 Hercules Stingray ALG-2228/2301/2302
 193 Hercules Stingray 64/V with ICS5342 ARK2000MT
 194 Hercules Stingray 64/V with ZoomDAC ARK1000PV
 195 Hercules Stingray Pro ARK1000PV
 196 Hercules Stingray Pro/V ARK1000PV
 197 Hercules Terminator 3D/DX S3 ViRGE/DX

Enter a number to choose the corresponding card definition.
 Press Enter for the next page; press q to continue configuration.

198 Hercules Terminator 64/3D S3 ViRGE
 199 Hercules Terminator 64/Video S3 Trio64V+
 200 Integral FlashPoint ET4000/W32(i/p)
 201 Intel 5430 CL-GD5430
 202 Interay PMC Viper ET6000
 203 JAX 8241 S3 801/805
 204 Jatton Video-58P ET6000
 205 Jatton Video-70P CL-GD5464
 206 Jazz Multimedia G-Force 128 ET6000
 207 LeadTek WinFast 3D S600 S3 ViRGE
 208 LeadTek WinFast S200 ET4000/W32(i/p)
 209 LeadTek WinFast S430 S3 968
 210 LeadTek WinFast S510 S3 968
 211 MELCO WGP-VG4S S3 ViRGE
 212 MELCO WGP-VX8 S3 ViRGE/VX
 213 Matrox Comet ET4000/W32(i/p)
 214 Matrox Marvel II ET4000/W32(i/p)
 215 Matrox Millennium (MGA) mga2064w

Enter a number to choose the corresponding card definition.
 Press Enter for the next page; press q to continue configuration.

216 Matrox Millennium II mga2164w
 217 Matrox Mystique mga1064sg
 218 MediaVision Proaxcel 128 ET6000
 219 Mirage Z-128 ET6000
 220 Miro Crystal 10SD with GenDAC S3 801/805
 221 Miro Crystal 12SD S3 Trio32
 222 Miro Crystal 16S S3 928
 223 Miro Crystal 20SD PCI with S3 SDAC S3 868
 224 Miro Crystal 20SD VLB with S3 SDAC (BIOS 3.xx) S3 864
 225 Miro Crystal 20SD with ICD2061A (BIOS 2.xx) S3 864
 226 Miro Crystal 20SD with ICS2494 (BIOS 1.xx) S3 864
 227 Miro Crystal 20SV S3 964
 228 Miro Crystal 22SD S3 Trio64
 229 Miro Crystal 40SV S3 964
 230 Miro Crystal 80SV S3 968
 231 Miro Crystal 8S S3 801/805
 232 Miro MiroVideo 20TD ET4000/W32(i/p)
 233 Miro Video 20SV S3 968

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

234 NeoMagic (laptop/notebook)	MagicGraph 128 series
235 Number Nine FX Motion 331	S3 Trio64V+
236 Number Nine FX Motion 332	S3 ViRGE
237 Number Nine FX Motion 531	S3 868
238 Number Nine FX Motion 771	S3 968
239 Number Nine FX Vision 330	S3 Trio64
240 Number Nine GXE Level 10/11/12	S3 928
241 Number Nine GXE Level 14/16	S3 928
242 Number Nine GXE64	S3 864
243 Number Nine GXE64 Pro	S3 964
244 Number Nine GXE64 with S3 Trio64	S3 Trio64
245 Number Nine Imagine I-128 (2-8MB)	I128
246 Number Nine Imagine I-128 Series 2 (2-4MB)	I128
247 Number Nine Visual 9FX Reality 332	S3 ViRGE
248 Oak (generic)	Oak-067/77/87
249 Ocean (octek) VL-VGA-1000	ARK1000VL
250 Orchid Celsius (AT&T RAMDAC)	AGX-015
251 Orchid Celsius (Sierra RAMDAC)	AGX-015

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

252 Orchid Fahrenheit 1280	S3 801
253 Orchid Fahrenheit VA	S3 801/805
254 Orchid Fahrenheit-1280+	S3 801/805
255 Orchid Kelvin 64	CL-GD5434
256 Orchid Kelvin 64 VLB Rev A	CL-GD5434
257 Orchid Kelvin 64 VLB Rev B	CL-GD5434
258 Orchid P9000 VLB	Weitek 9000
259 Orchid Technology Fahrenheit Video 3D	S3 ViRGE
260 Paradise Accelerator Value	Oak OTI-087
261 Paradise/WD 90CXX	WD90CXX
262 Rendition Verite 1000	Verite 1000
263 S3 801/805 (generic)	S3 801/805
264 S3 801/805 with ATT20c490 RAMDAC	S3 801/805
265 S3 801/805 with ATT20c490 RAMDAC and ICD2061A	S3 801/805
266 S3 801/805 with Chronitel 8391	S3 801/805
267 S3 801/805 with S3 GenDAC	S3 801/805
268 S3 801/805 with SC1148{2,3,4} RAMDAC	S3 801/805
269 S3 801/805 with SC1148{5,7,9} RAMDAC	S3 801/805

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

270 S3 864 (generic)	S3 864
271 S3 864 with ATT 20C498 or 21C498	S3 864
272 S3 864 with SDAC (86C716)	S3 864
273 S3 864 with STG1703	S3 864
274 S3 868 (generic)	S3 868
275 S3 868 with ATT 20C409	S3 868
276 S3 868 with ATT 20C498 or 21C498	S3 868
277 S3 868 with SDAC (86C716)	S3 868
278 S3 86C325 (generic)	S3 ViRGE
279 S3 86C375 (generic)	S3 ViRGE/DX
280 S3 86C385 (generic)	S3 ViRGE/GX
281 S3 86C764 (generic)	S3 Trio64
282 S3 86C765 (generic)	S3 Trio64V+
283 S3 86C775 (generic)	S3 Trio64V2

284 S3 86C785 (generic)	S3 Trio64V2
285 S3 86C801 (generic)	S3 801/805
286 S3 86C805 (generic)	S3 801/805
287 S3 86C864 (generic)	S3 864

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

288 S3 86C868 (generic)	S3 868
289 S3 86C911 (generic)	S3 911/924
290 S3 86C924 (generic)	S3 911/924
291 S3 86C928 (generic)	S3 928
292 S3 86C964 (generic)	S3 964
293 S3 86C968 (generic)	S3 968
294 S3 86C988 (generic)	S3 ViRGE/VX
295 S3 911/924 (generic)	S3 911/924
296 S3 924 with SC1148 DAC	S3 924
297 S3 928 (generic)	S3 928
298 S3 964 (generic)	S3 964
299 S3 968 (generic)	S3 968
300 S3 Trio32 (generic)	S3 Trio32
301 S3 Trio64 (generic)	S3 Trio64
302 S3 Trio64V+ (generic)	S3 Trio64V+
303 S3 Trio64V2 (generic)	S3 Trio64V2
304 S3 Trio64V2/DX (generic)	S3 Trio64V2
305 S3 Trio64V2/GX (generic)	S3 Trio64V2

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

306 S3 ViRGE (S3V server)	S3 ViRGE
307 S3 ViRGE (generic)	S3 ViRGE
308 S3 ViRGE/DX (generic)	S3 ViRGE/DX
309 S3 ViRGE/GX (generic)	S3 ViRGE/GX
310 S3 ViRGE/VX (generic)	S3 ViRGE/VX
311 S3 Vision864 (generic)	S3 864
312 S3 Vision868 (generic)	S3 868
313 S3 Vision964 (generic)	S3 964
314 S3 Vision968 (generic)	S3 968
315 SNI PC5H W32	ET4000/W32(i/p)
316 SNI Scenic W32	ET4000/W32(i/p)
317 SPEA Mercury 64	S3 964
318 SPEA Mirage	S3 801/805
319 SPEA/V7 Mercury	S3 928
320 SPEA/V7 Mirage P64	S3 864
321 SPEA/V7 Mirage P64 with S3 Trio64	S3 Trio64
322 SPEA/V7 Mirage VEGA Plus	ALG-2228
323 SPEA/V7 ShowTime Plus	ET4000/W32(i/p)

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press q to continue configuration.

324 STB Horizon	CL-GD5426/28
325 STB Horizon Video	CL-GD5440
326 STB LightSpeed	ET4000/W32(i/p)
327 STB LightSpeed 128	ET6000
328 STB MVP-2	ET4000
329 STB MVP-2 PCI	ET4000/W32(i/p)
330 STB MVP-2X	ET4000/W32(i/p)
331 STB MVP-4 PCI	ET4000/W32(i/p)
332 STB MVP-4X	ET4000/W32(i/p)

333 STB Nitro (64)	CL-GD5434
334 STB Nitro 30	S3 ViRGE/GX
335 STB Nitro 64 Video	CL-GD5446
336 STB Pegasus	S3 928
337 STB Powergraph 64	S3 Trio64
338 STB Powergraph 64 Video	S3 Trio64V+
339 STB Powergraph X-24	S3 801/805
340 STB Systems Powergraph 3D	S3 ViRGE
341 STB Systems Velocity 3D	S3 ViRGE/VX

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press **q** to continue configuration.

342 STB Velocity 64 Video	S3 968
343 SiS SG86C201	SIS86C201
344 Sierra Screaming 3D	Verite 1000
345 Sigma Concorde	ET4000/W32
346 Sigma Legend	ET4000
347 Spider Black Widow	AGX-015
348 Spider Black Widow Plus	AGX-016
349 Spider Tarantula 64	S3 964
350 Spider VLB Plus	CL-GD5428
351 TechWorks Thunderbolt	ET4000/W32
352 TechWorks Ultimate 3D	CL-GD5464
353 Trident 8900/9000 (generic)	TVGA8900/9000
354 Trident 8900D (generic)	TVGA8900D
355 Trident TGUI9400CXi (generic)	TGUI9400CXi
356 Trident TGUI9420DGi (generic)	TGUI9420DGi
357 Trident TGUI9430DGi (generic)	TGUI9430DGi
358 Trident TGUI9440 (generic)	TGUI9440
359 Trident TGUI9660 (generic)	TGUI9660

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press **q** to continue configuration.

360 Trident TGUI9680 (generic)	TGUI9680
361 Trident TVGA9200Cxr (generic)	TVGA9200Cxr
362 Unsupported VGA compatible	Generic VGA
363 VI720	CL-GD5434
364 VL-41	S3 801/805
365 VidTech FastMax P20	S3 864
366 VideoLogic GrafixStar 300	S3 Trio64
367 VideoLogic GrafixStar 400	S3 Trio64V+
368 VideoLogic GrafixStar 500	S3 868
369 VideoLogic GrafixStar 550	CL-GD5464
370 VideoLogic GrafixStar 600	ET6000
371 VideoLogic GrafixStar 700	S3 968
372 ViewTop PCI	ET4000/W32(i/p)
373 WD 90C24 (laptop)	WD90C24
374 WD 90C24A or 90C24A2 (laptop)	WD90C24A
375 WinFast 3D S600	S3 ViRGE
376 WinFast S200	ET4000/W32(i/p)
377 WinFast S430	S3 968

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press **q** to continue configuration.

378 WinFast S510	S3 968
379 XGA-1 (ISA bus)	XGA-1
380 XGA-2 (ISA bus)	XGA-2
381 miro miroMedia 3D	S3 ViRGE

Enter a number to choose the corresponding card definition.
Press Enter for the next page; press **q** to continue configuration.

Selectați placa video tastând numărul corespunzător acesteia. De exemplu, dacă PC-ul dumneavoastră are o placă video Number Nine GXE64 cu cipset de tip S3-Trio64, trebuie să tastați **244** și apoi să apăsați Enter.



Sugestie

Fiecare nouă variantă a sistemului XFree86 suportă din ce în ce mai multe plăci. Lista din această secțiune prezintă plăcile suportate de varianta XFree86 3.3.1 (pentru comparație, în versiunea 3.1.2, lista conținea doar 124 de plăci, față de 381 în prezent). Dacă modelul și marca plăcii dumneavoastră nu apar în această listă, dar cunoașteți cipsetul video, încercați să selectați placa din listă care corespunde cipsetului generic. De exemplu, dacă placa dumneavoastră utilizează un cipset de tip S3 Trio64, dar nu îi cunoașteți marca și modelul, selectați din listă placa numerotată cu 301.

După ce ați selectat placa, `xf86config` afișează anumite informații (și instrucțiuni corespunzătoare) referitoare la placa selectată. La selectarea plăcii cu numărul 244 spre exemplu, `xf86config` va afișa următoarele informații:

Your selected card definition:

```
Identifier: Number Nine GXE64 with S3-Trio64
Chipset: S3-Trio64
Server: XF86_S3
Do NOT probe clocks or use any Clocks line.
```

Press Enter to continue or ctrl-c to abort.



Sugestie

Remarcați recomandarea de a nu verifica semnalele de ceas și de a nu utiliza o linie Clocks. Aceste recomandări trebuie respectate în configurarea ulterioară a serverului cu ajutorul programului `xf86config`. Dacă programul vă spune să verificați semnalele de ceas, trebuie însă să răspundeți cu No. Nu aveți nevoie să știți ce se înțelege prin „verificarea semnalelor de ceas”. Tot ce trebuie este să vă conformați recomandării, pentru a obține un fișier `XF86Config` care să poată fi utilizat corect.

După ce ați apăsat Enter pentru a continua, `xf86config` va afișa următorul mesaj, cerându-vă să selectați un anumit server X:

Now you must determine which server to run. Refer to the manpages and other documentation. The following servers are available (they may not all be installed on your system):

- 1 The XF86_Mono server. This is a monochrome server that should work on any VGA-compatible card, in 640x480 (more on some SVGA chipsets).
- 2 The XF86_VGA16 server. This is a 16-color VGA server that should work on any VGA-compatible card.
- 3 The XF86_SVGA server. This is a 256 color SVGA server that supports a number of SVGA chipsets. On some chipsets it is accelerated or supports higher color depths.
- 4 The accelerated servers. These include XF86_S3, XF86_Mach32, XF86_Mach8, XF86_8514, XF86_P9000, XF86_AGX, XF86_W32, XF86_Mach64, XF86_I128 and XF86_S3V.

These four server types correspond to the four different "Screen" sections in `XF86Config` (`vga2`, `vga16`, `svga`, `accel`).

5 Choose the server from the card definition, `XF86_S3`.

Which one of these screen types do you intend to run by default (1-5)? 5

Dacă ați precizat corect placa video, puteți apăsa **5** pentru a selecta serverul corespunzător acesteia. Programul `xf86config` va afișa în continuare următorul mesaj, urmat de o întrebare:

The server to run is selected by changing the symbolic link 'X'. For example, 'rm /usr/X11R6/bin/X; ln -s /usr/X11R6/bin/XF86_SVGA /usr/X11R6/bin/X' selects the SVGA server.

Please answer the following question with either 'y' or 'n'.
Do you want me to set the symbolic link?

Apăsați **y** pentru a răspunde afirmativ la această întrebare.

În continuare, xf86config vă cere să precizați de câtă memorie dispune placa video:

How much video memory do you have on your video card:

- 1 256K
- 2 512K
- 3 1024K
- 4 2048K
- 5 4096K
- 6 Other

Majoritatea plăcilor utilizate în prezent au între 2 MB (2048 KB) și 4 MB (4096 KB) de memorie video. Aveți nevoie de minimum 1 MB de memorie video pentru a putea afișa 256 de culori la o rezoluție de 1024x768 (adică 1024 de pixeli pe orizontală și 768 de pe verticală). Introduceți numărul care corespunde cantității de memorie disponibilă pe placa video și apăsați apoi Enter.

Trebuie în continuare să introduceți o informație de identificare a plăcii video. Introduceți informația corespunzătoare plăcii dumneavoastră. Iată un exemplu:

Enter an identifier for your video card definition: **Number Nine GXE64**
Enter the vendor name of your video card: **Number Nine**
Enter the model (board) name of your video card: **GXE64**

Programul xf86config vă cere în ceea ce urmează câteva informații de natură tehnică utile în special pentru serverele accelerate. Mai întâi, programul vă cere parametrul RAMDAC: apăsați **q** și apoi Enter pentru a continua. Programul vă cere apoi să precizați tipul circuitului de ceas. Dacă nu cunoașteți această informație, apăsați Enter pentru a continua.

Urmează linia Clocks, care reprezintă o listă a frecvențelor de sincronizare pe care serverul X la utilizează atunci când se lansează. Fără această linie, serverul X va determina frecvențele de sincronizare la fiecare lansare. Serverul X poate rula în *mod de testare* (dacă este pornit cu linia X -probeonly), mod în care serverul determină și afișează frecvențele de sincronizare. Puteți introduce ulterior această informație în fișierul XF86Config. De fapt, programul xf86config poate mai întâi să ruleze serverul X în mod de testare pentru a determina valorile acestor frecvențe.

Înainte de a continua, verificați dacă placa video are nevoie de o linie Clocks (amintiți-vă de recomandările afișate de xf86config după selectarea plăcii video). Dacă veți preciza o placă de tip S3, va apărea mesajul:

The card definition says NOT to probe clocks.
Do you want me to run 'X -probeonly' now ? **N**

Desigur, în acest caz trebuie să răspundeți negativ. Pe de altă parte, dacă placa poate utiliza o linie Clocks, apăsați **y** pentru a răspunde afirmativ. În acest caz, programul xf86config va rula serverul X în modul de testare, va determina valorile de sincronizare și va adăuga în fișierul de configurare linia Clocks corespunzătoare.

În continuare, programul xf86config afișează mai multe moduri și vă cere să selectați unul dintre acestea, ca în exemplul:

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed mode that can be supported by the monitor and card.

Currently it is set to:

"640x480" "800x600" "1024x768" for 8bpp
"640x480" "800x600" "1024x768" for 16bpp
"640x480" for 24bpp
"640x480" for 32bpp

Note that 16, 24 and 32bpp are only supported on a few configurations. Modes that cannot be supported due to monitor or clock constraints will be automatically skipped by the server.

- 1 Change the modes for 8pp (256 colors)
- 2 Change the modes for 16bpp (32K/64K colors)
- 3 Change the modes for 24bpp (24-bit color, packed pixel)
- 4 Change the modes for 32bpp (24-bit color)
- 5 The modes are OK, continue.

Enter your choice: **5**

Prescurtarea bpp vine de la „*biți pe pixel*”. Adâncimea unui anumit mod se exprimă prin valori cum sunt 8 bpp sau 16 bpp. Puteți alege ultima opțiune din listă, continuând apoi configurarea. Tastați deci **5** și apoi Enter pentru a accepta aceste opțiuni.

Acest pas încheie sesiunea de lucru cu xf86config. Programul va afișa următorul mesaj, întrebându-vă dacă poate scrie fișierul XF86Config în directorul curent:

I am going to write the XF86Config file now. Make sure you don't accidentally overwrite a previously configured one.

Do you want it written to the current drectory as 'XF86Config' ? **n**

Please give a filename to write to: **/etc/X11/XF86Config**

Răspundeți cu **n** la prima întrebare, după care tastați **/etc/X11/XF86Config** și apăsați Enter. Programul xf86config va scrie fișierul de configurare și se va termina.

În cazul fericit, fișierul XF86Config generat de xf86config este suficient pentru ca serverul X să ruleze pe configurația dumneavoastră. Din cauza posibilității de defectare a monitorului din cauza unor valori greșite în fișierul XF86Config, se recomandă să utilizați un editor de texte pentru a examina acest fișier (capitolul 24, „Prelucrarea textului în Linux”, descrie editoarele vi și GNU Emacs).

Verificarea fișierului XF86Config

În urma execuției programului xf86Config, se creează fișierul XF86Config; are totodată loc selecția unui server X corespunzător plăcii video. Fișierul XF86Config are rolul de a descrie serverului X placa video, monitorul și mouse-ul dumneavoastră. În mod implicit, serverul X caută mai întâi fișierul de configurare în directorul /etc/X11 (/etc/X11/XF86Config), loc în care se recomandă să salvați și dumneavoastră fișierul de configurare creat.

Analizând fișierul XF86Config oferit ca model, puteți constata că acesta este compus din mai multe secțiuni. Fiecare secțiune are următorul format:

```
# Acesta este un comentariu
Section „NumeSecțiune“
    NumeElement ValoareElement
    ...
    Subsection „NumeSubsecțiune“
        NumeElement ValoareElement
```

```
...
EndSubsection
EndSection
```

Fiecare secțiune este compusă dintr-o secvență de elemente; fiecare dintre aceste elemente are un nume și o valoare. O secțiune poate conține una sau mai multe subsecțiuni. Linii de comentariu sunt marcate prin intermediul semnului „diez” (#) la începutul unei linii.

Fișierul `XF86Config` conține una sau mai multe dintre următoarele șapte secțiuni:

- **Files** (Fișiere). Această secțiune cuprinde o listă a căilor (numele complete de directoare) fișierelor cu fonturi și fișierului care conține baza de date cu culori, numit *fișier RGB*. Abrevierea RGB provine de la numele în limba engleză ale celor trei culori fundamentale – roșu, verde și albastru.

- **ServerFlags** (Opțiuni ale serverului X). Această secțiune conține diferite opțiuni ale serverului X, cum ar fi `DontZap` (ceea ce înseamnă „nu permite terminarea serverului X prin apăsarea combinației de taste `Ctrl+Alt+Backspace`”) sau `DontZoom` (care înseamnă „nu accepta modificarea rezoluției ecranului prin apăsarea unor combinații de taste”).

- **Keyboard** (Tastatură). În această secțiune sunt precizate tipul tastaturii și anumite caracteristici ale acesteia, cum ar fi durata și viteza de repetare.

- **Pointer** (Indicator). Această secțiune cuprinde informații referitoare la mouse, inclusiv numele dispozitivului asociat acestuia. (În terminologia sistemului X Window, mouse-ul este denumit *indicator*.)

- **Monitor** (Monitor). Această secțiune include parametrii monitorului (cum ar fi ratele de sincronizare pe orizontală și pe verticală) și o listă de moduri video pe care monitorul le suporta.

- **Device** (Placa video). În această secțiune, sunt prezentate caracteristicile plăcii video (adaptorului grafic). Fișierul de configurare poate conține mai multe secțiuni `Device`.

- **Screen** (Ecran). Această secțiune prezintă combinația caracteristicilor plăcii video și monitorului utilizate de serverul X. De regulă, fișierul de configurare are mai multe secțiuni `Screen`.

În majoritatea situațiilor, nu trebuie să învățați toate detaliile legate de aceste secțiuni; fișierul `XF86Config` trebuie să reprezinte punctul dumneavoastră de pornire. Următoarele câteva secțiuni rezumă câteva dintre cele mai importante secțiuni din fișierul de configurare și vă îndrumă asupra modificărilor pe care trebuie să le efectuați.

Secțiunea Files

Secțiunea `Files` cuprinde locațiile unora dintre fișierele de care serverul X are nevoie. Acestea sunt următoarele:

- Un fișier care conține definițiile culorilor (acesta este numit *fișier RGB*)
- Locațiile fonturilor (fiecare astfel de locație este un director separat)

Nu trebuie să modificați numele fișierului RGB. În ceea ce privește locațiile fonturilor, asigurați-vă că fiecare dintre aceste directoare există. Acestea ar trebui să existe, deoarece toate fonturile au fost instalate la instalarea sistemului Linux. Dacă unul dintre directoare nu există, trebuie să ștergeți linia corespunzătoare din fișierul de configurare sau să tastați # la începutul liniei pentru a o comenta.

Secțiunea ServerFlags

În această secțiune, poate apărea oricare dintre opțiunile prezentate în tabelul 4.1; pentru ca o anumită opțiune să fie activă, este necesar să o scrieți pe o linie separată în cadrul fișierului de configurare.

Tabelul 4-1 Opțiunile din secțiunea ServerFlags

Opțiunea	Semnificația
<code>NoTrapSignals</code>	Se utilizează numai pentru depanare.
<code>DontZap</code>	Serverul X va ignora secvența <code>Ctrl+Alt+Backspace</code> , care determină de regulă terminarea serverului.
<code>DontZoom</code>	Serverul X va ignora secvențele <code>Ctrl+Alt+Keypad+</code> și <code>Ctrl+Alt+Keypad-</code> , utilizate în mod obișnuit pentru schimbarea modurilor video. (<i>Keypad+</i> și <i>Keypad-</i> reprezintă tastele plus și minus de pe tastatura numerică.)

Fișierul de configurare generat de programul `xf86config` nu activează nici una dintre aceste opțiuni; prin urmare, se recomandă să nu modificați această secțiune. Mai târziu, combinația `Ctrl+Alt+Backspace` va fi utilă atunci când serverul nu va funcționa sau se va bloca. Combinațiile `Ctrl+Alt+Keypad+` și `Ctrl+Alt+Keypad-` sunt utile atunci când doriți să încercați mai multe moduri video fără să reporniți serverul X.

Secțiunea Monitor

Secțiunea `Monitor` este esențială în cadrul fișierului `XF86Config`, deoarece cuprinde anumiți parametri tehnici importanți ai monitorului. Dacă ați cumpărat de curând calculatorul, probabil că dispuneți de un manual al monitorului. În acest manual, veți găsi o pagină de specificații tehnice. Informațiile de pe acea pagină vă sunt necesare pentru completarea secțiunii `Monitor`.

Asigurați-vă că secțiunea conține un element `Identifier`, ca în exemplul:

```
Identifier "Dell VS15X"
```

Secțiunea `Screen`, prezentată ulterior, se va referi la monitor prin intermediul numelui `Dell VS15X`.

Următorii doi parametri – `HorizSync` și `VertRefresh` – sunt extrem de importanți. Parametrul `HorizSync` precizează domeniul de frecvențe la care se efectuează sincronizarea pe orizontală. Introduceți acest domeniu exact așa cum este specificat în manualul monitorului. Frecvențele pentru sincronizarea pe orizontală se măsoară de regulă în kilohertzi (kHz).

Cât despre `VertRefresh`, trebuie să introduceți domeniul de frecvențe de reîmprospătare pe verticală (denumite și frecvențe de sincronizare pe verticală în manualele anumitor monitoare). Aceste frecvențe se măsoară în hertzi (Hz).

Pentru monitorul `Dell VS15X`, parametrii `HorizSync` și `VertRefresh` (obținuți din manualul monitorului) au următoarele valori:

```
# NOTA: ACESTE VALORI AU NUMAI ROL DE EXEMPLU.
# INTRODUCETI VALORILE DIN MANUALUL MONITORULUI DUMNEAVOASTRA
HorizSync 30-64
VertRefresh 55-90
```

După aceste valori critice, urmează un număr mare de poziții `Modeline`. Această listă reprezintă setul modurilor grafice standard; serverul X va alege în mod implicit din acest set pe cel mai bun. Nu trebuie să modificați aceste linii.

Secțiunea Device

Secțiunea `Device` precizează placa video, căreia îi asociază un nume (prin intermediul unei linii `Identifier`). Programul `xf86config` generează un fișier de configurare care conține secțiunea `Device` standard pentru o placă VGA. Trebuie să comparați secțiunea `Device` din fișier cu

datele pe care le-ați introdus atunci când ați rulat `xf86config`. Linia Identifier trebuie să conțină identificatorul corect al plăcii video, ca în exemplul:

Identifier "Number Nine GXE64 with S3-Trio64"

În secțiunea Screen, placa video va fi referită prin intermediul numelui din secțiunea Identifier. Pentru majoritatea plăcilor video, această secțiune nu mai conține și alte informații, deoarece serverul X (asociat acelei plăci) poate configura și utiliza corect placa.



Sugestie

Pentru anumite plăci video, secțiunea Device din fișierul `XF86Config` trebuie să conțină un parametru Clocks, a cărui valoare este reprezentată printr-o secvență de numere fracționare. Aceste numere sunt destul de dificil de determinat (deși anumite cipseturi video, printre care și S3-Trio64, nu au nevoie de ele). Un mod de a le determina este prin rularea serverului X cu comanda `X -probeonly`. De fapt, programul `xf86config` rulează serverul X cu această comandă, determinând aceste valori. Este suficient să răspundeți cu `y` la întrebarea:

Do you want me to run 'X -probeonly' now?

Dacă placa video are nevoie de aceste valori în secțiunea Device, această metodă este cel mai simplu mod de a le determina.

Secțiunea Screen

Fiecare secțiune Screen specifică valorile Monitor și Device care vor fi utilizate de serverul X. Acești parametri sunt specificați prin numele prezentate în pozițiile Identifier corespunzătoare. Serverul X este identificat printr-un element Driver.

Programul `xf86config` generează mai multe secțiuni Screen în fișierul de configurare. Fiecare dintre aceste secțiuni este destinată unui server X diferit, identificat prin elementul Driver. Opțiunile posibile sunt prezentate în tabelul 4-2.

Tabelul 4-2 Elementele Driver

Poziția	Scopul
accel	Server X pentru cipseturi video accelerate (S3, Mach32, Mach64, P9000, 8514, AGX sau W32)
svga	Server X pentru plăci SuperVGA
vga16	Server X pentru plăci generice VGA cu 16 culori
vga2	Server X pentru plăci VGA monocrome

Fiecare secțiune Screen conține și câteva subsecțiuni Display, care indică modurile video (cum ar fi 640x480, pentru 640 de pixeli pe orizontală și 480 pe verticală; 800x600 sau 1024x768). Fiecare subsecțiune Display conține o poziție Depth care precizează numărul biților utilizați pentru fiecare pixel – număr care reprezintă adâncimea unui pixel.

Serverul X va selecta secțiunea Screen al cărui element Driver corespunde tipului serverului X (SuperVGA sau Accelerated spre exemplu). Acesta este precizat la rularea programului `xf86config`.

Rularea serverului X

După ce fișierul `XF86Config` este complet, sunteți gata de a porni serverul X și anumite aplicații X. Pentru aceasta, utilizați scriptul `startx`, care este un fișier de comenzi Linux. Acest script se găsește în directorul `/usr/X11/bin`, care trebuie să fie inclus în variabila de mediu PATH. Pentru a rula acest script, tastați `startx` în linia de comandă.

Scriptul `startx` caută un alt fișier de comenzi, `.xinitrc`, în directorul dumneavoastră personal. Dacă `startx` nu găsește fișierul `.xinitrc` în directorul dumneavoastră personal, rulează comanda `xinit` cu scriptul implicit `/usr/X11R6/lib/X11/xinit/xinitrc`. Observați că spre deosebire de fișierul `.xinitrc` din directorul dumneavoastră personal, numele scriptului prestabilit nu începe cu punct.

Rezultatul rulării comenzii `startx` depinde de comenzile din fișierul `.xinitrc` din directorul dumneavoastră personal, dacă acest fișier există. În caz contrar, rezultatul depinde de comenzile din fișierul `/usr/X11R6/lib/X11/xinit/xinitrc`. Cu versiunea implicită a acestui fișier, la rularea scriptului `startx` ca utilizator obișnuit (nu root), ecranul dumneavoastră trebuie să arate similar cu cel din figura 4-1.

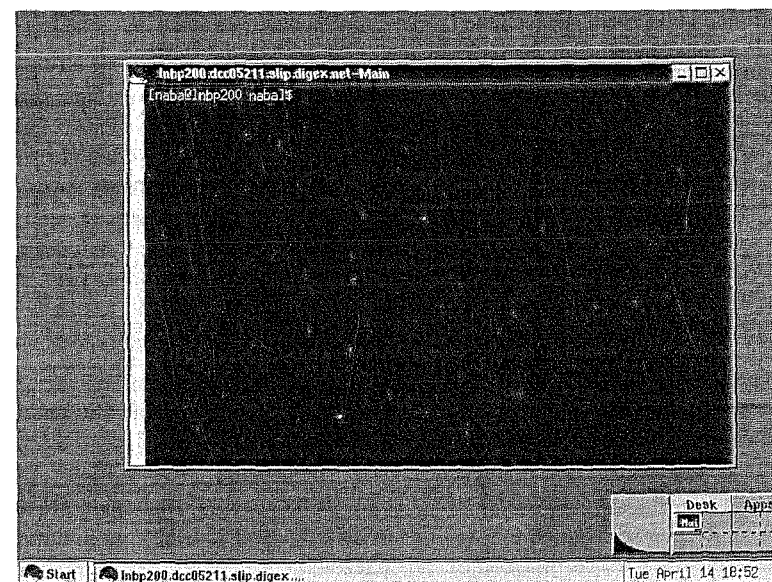


Referință încrucișată

Puteți introduce acum comenzi Linux la promptul ferestrei `xterm` (fereastra mare din mijlocul ecranului). Veți afla mai multe despre personalizarea mediului X în capitolul 5.

Figura 4-1

Ecranul X inițial



Sugestie

Serverul X se termină la terminarea scriptului `startx`. Scriptul `startx`, pe de altă parte, se va termina la terminarea ultimei comenzi pe care o conține. De regulă, programul de gestiune a ferestrelor (ca de exemplu `fvwm2`) reprezintă ultima comandă – deci puteți opri serverul X, întorcându-vă la ecranul text, părăsind programul de gestiune a ferestrelor.

Părăsirea sistemului utilizând combinația Ctrl+Alt+Backspace

Dacă monitorul și placa video nu sunt corect specificate în fișierul `XF86Config`, este foarte probabil ca la pornirea serverului X cu comanda `startx`, să nu apară ecranul X prezentat în figura 4-1. Dacă întâmpinați probleme cum ar fi un ecran distorsionat sau sistemul se

blochează, apăsați combinația Ctrl+Alt+Backspace pentru a „ucide” serverul X și a vă întoarce în modul text. (În UNIX, termenul de „a ucide” se referă la terminarea forțată a unui program. În UNIX există chiar o comandă kill care se utilizează pentru a opri programele eronate.)

După oprirea serverului X, puteți edita din nou fișierul XF86Config pentru a determina ce este eronat.

Încercarea diferitelor moduri video

În fișierul XF86Config, dacă veți examina secțiunea Screen corespunzătoare serverului X pe care îl rulați, veți observa că apar mai multe subsecțiuni Display. Fiecare subsecțiune Display cuprinde modurile video suportate pentru o anumită valoare a adâncimii – numărul de biți prin care se reprezintă valoarea fiecărui pixel. Serverele X suportă de regulă adâncimea de 8, ceea ce înseamnă că fiecare pixel are o valoare pe 8 biți, iar serverul poate afișa simultan $2^8=256$ culori distincte. Fiecare subsecțiune Display descrie un anumit mod video în termenii rezoluției de afișare, care, la rândul său, se exprimă prin numărul de pixeli de pe orizontală și respectiv de pe verticală.

De exemplu, secțiunea Screen pentru un server X accelerat utilizat de o placă video S3-Trio64, cuprinde următoarea subsecțiune Display corespunzătoare adâncimii de 8 biți pe pixel:

```
# The accelerated servers (S3, Mach32, Mach8, 8514, P9000,
# AGX, W32, Mach64)
Section "Screen"
    Driver      "accel"
    Device      "Number Nine GXE64 with S3-Trio64"
    Monitor     "Dell VS15X"
    Subsection "Display"
        Depth      8
        Modes       "640x480" "800x600" "1024x768"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection
```

...

La pornirea serverului X, acesta configurează placa video la rezoluția (în acest caz, 640x480 – adică 640 de pixeli pe orizontală și 480 pe verticală) corespunzătoare primului dintre modurile listate în linia Modes, ca în exemplul:

```
Modes      "640x480" "800x600" "1024x768"
```

Puteți încerca și alte moduri fără a părăsi serverul X. Apăsați combinația Ctrl+Alt+Keypad+ (Keypad+ reprezintă tasta plus de pe tastatura extinsă). Serverul X va comuta la modul video care urmează în listă – în acest caz, cel cu rezoluția de 800x600. Apăsând din nou combinația Ctrl+Alt+Keypad+, serverul va trece în modul 1024x768. Prin urmare, la apăsarea combinației Ctrl+Alt+Keypad+, serverul X va alege următorul mod din lista existentă în linia Modes.



Apăsați de câteva ori combinația Ctrl+Alt+Keypad+, pentru a vă asigura că serverul X funcționează corect în toate modurile video.

Sugestie

Pentru a vă deplasa înapoi la modul precedent, utilizați combinația Ctrl+Alt+Keypad- (prin Keypad-, înțelegem tasta minus de pe tastatura numerică). De exemplu, dacă serverul X afișează în mod curent în modul 800x600, după apăsarea combinației Ctrl+Alt+Keypad-, acesta va trece la modul 640x480.

Puteți determina serverul X să pornească în oricare dintre modurile pe care acesta le suportă. Dacă doriți ca serverul să pornească în modul cu rezoluția maximă, este suficient să modificați

linia Modes din secțiunea Screen corespunzătoare serverului X (tipul acestuia este indicat de linia Driver din secțiunea Screen) după cum urmează:

```
Modes      "1024x768" "800x600" "640x480"
```

Această modificare determină pornirea serverului în modul cu rezoluția 1024x768, spațiul pe ecran fiind considerabil mai mare decât în modul 640x480.

Rezoluțiile din linia Modes determină următoarele:

- Prima rezoluție din listă este cea prestabilită (rezoluția la care pornește serverul X).
- La modificarea rezoluției ecranului, serverul X alege noua rezoluție în ordinea indicată în linia Modes. La apăsarea combinației Ctrl+Alt+Keypad+, serverul X va modifica rezoluția parcurgând lista de la stânga spre dreapta. Ordinea se inversează dacă se utilizează combinația Ctrl+Alt+Keypad-.

Părăsirea programului de gestiune a ferestrelor

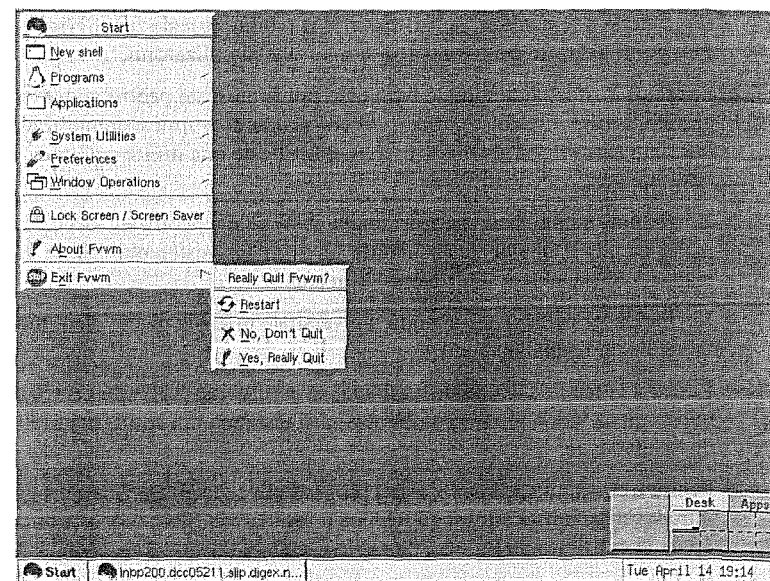
Pentru a părăsi ecranul X și a vă întoarce în mod text, trebuie să părăsiți ultimul dintre programele pe care scriptul startx le pornește. Acesta este programul de gestiune a ferestrelor – în majoritatea cazurilor, programul fvwm2.

Pentru a părăsi programul fvwm2, deplasați cursorul mouse-ului într-o zonă liberă de pe ecran și executați clic-stânga. Programul de gestiune a ferestrelor va afișa un meniu. Câteva dintre opțiunile acestui meniu au săgeți care indică spre dreapta. Deplasând cursorul mouse-ului pe una dintre aceste săgeți, va apărea un meniu în cascadă.

Deplasați cursorul mouse-ului pe opțiunea numită Exit fvwm, care are o săgeată care indică spre dreapta. Programul de gestiune a ferestrelor va afișa un submeniu, după cum se observă și în figura 4-2.

Figura 4-2

Părăsirea
programului de
gestiune a
ferestrelor fvwm2



Deplasați cursorul pe opțiunea numită Yes, Really Quit și eliberați butonul mouse-ului. Programul `fwm2` se va termina; aceasta va cauza terminarea scriptului `startx` și prin urmare și a serverului X. Ecranul va reveni la modul text, afișând promptul de comandă al sistemului Linux.

Rezumat

Sistemul X Window, pe scurt X, reprezintă un sistem de ferestre care servește ca bază a interfețelor grafice și totodată ca mediu grafic pe majoritatea stațiilor UNIX. Implementarea sistemului X pentru calculatoare PC cu procesoare Intel 80x86 și compatibile poartă numele de XFree86 și este disponibilă în mod gratuit. Sistemul XFree86 poate lucra cu un număr mare de plăci video, dar trebuie configurat astfel încât să utilizeze parametri adecvați pentru placa video și monitorul din configurația dumneavoastră. Acest capitol prezintă modul în care puteți configura și rula X pe sistemul dumneavoastră Linux. În capitolul următor, ne vom concentra asupra personalizării și utilizării interfeței grafice oferite de X și de programul de gestiune a ferestrelor.

După lectura acestui capitol, trebuie să rețineți următoarele:

- ▶ Sistemul X Window reprezintă un sistem de ferestre transparent în rețea, având la bază modelul client/server. Serverul X, rulând pe o stație de lucru cu un sistem de afișare de tip „hartă de biți”, gestionează zone ale ecranului denumite ferestre, în care aplicațiile client X își pot afișa rezultatele. Clienții X rulează adeseori pe sisteme aflate la distanță, chiar dacă rezultatele lor apar pe ecranul curent.
- ▶ Termenul de interfață grafică (GUI) descrie o interfață pentru utilizatori care utilizează ferestre, meniuri și alte obiecte grafice, astfel încât utilizatorii să poată interacționa cu aplicația prin deplasarea și apăsarea butoanelor mouse-ului. Din punctul de vedere al proiectantului de aplicații, o interfață grafică reprezintă o combinație între un program de gestiune a ferestrelor, un stil specific și o bibliotecă de funcții (set de instrumente) care pot fi utilizate pentru a construi interfața cu utilizatorul.
- ▶ Sistemul X oferă funcțiile de bază care pot fi utilizate pentru construirea unei interfețe grafice. Multe interfețe grafice sunt construite deasupra sistemului X. Două astfel de interfețe sunt Motif și OPEN LOOK. Motif este una dintre cele mai răspândite interfețe grafice.
- ▶ XFree86 este implementarea sistemului X Window pentru PC-uri cu Linux și este inclus pe CD-ROM-ul atașat lucrării. La instalarea sistemului Linux de pe CD-ROM, realizați de asemenea și instalarea XFree86.
- ▶ Versiunea XFree86 3.3.1 suportă o diversitate mare de plăci video, inclusiv cele fabricate de firma Diamond.
- ▶ Pentru a putea rula X, trebuie mai întâi să configurați XFree86 pentru ca acesta să poată utiliza placa video, monitorul, mouse-ul și tastatura din configurația PC-ului dumneavoastră.
- ▶ Pentru a configura XFree86, rulați programul `xf86config`. Acest program vă cere să introduceți anumite informații tehnice referitoare la placa video, monitorul și mouse-ul aflate în configurația PC-ului dumneavoastră.
- ▶ Trebuie să precizați cât se poate de exact informațiile despre monitorul dumneavoastră, deoarece informațiile eronate pot conduce la nefuncționarea monitorului.

- ▶ Programul `xf86config` creează fișierul de configurare `/etc/X11/XF86Config`. Trebuie să citiți acest fișier înainte de a rula X.
- ▶ Pentru a porni serverul X, este suficient să tastați comanda `startx`. Pentru pornirea automată a unui set de aplicații X, trebuie să includeți comenzile care lansează acele aplicații într-un fișier denumit `.xinitrc`, care se găsește în directorul dumneavoastră personal.
- ▶ Dacă serverul X vă blochează sistemul, utilizați combinația `Ctrl+Alt+Backspace` pentru a ucidă serverul.
- ▶ Puteți schimba diferitele moduri video (cu rezoluții cum sunt 1024x768 sau 800x600), utilizând combinațiile speciale de taste `Ctrl+Alt+Keypad+` și `Ctrl+Alt+Keypad-`.

Capitolul 5

Personalizarea lansării sistemului Linux

În acest capitol:

- ▶ Veți înțelege etapele de lansare a sistemului Linux
- ▶ Veți obține un prompt grafic de intrare în sistem cu xdm
- ▶ Veți învăța să personalizați aplicațiile X prin intermediul resurselor
- ▶ Veți înțelege programul de gestionare a ferestrelor fwm2

În acest moment, ați instalat Linux și X și ați realizat primele teste ale sistemului Linux. Cu toate acestea, probabil că intrați în sistem folosind încă un ecran text, după care lansați sistemul X Window System manual. Dacă aveți experiență cu generarea curentă pe stațiile UNIX sau cu Microsoft Windows, sunteți familiarizat cu comoditatea în utilizare a unei interfețe cu utilizatorul. De asemenea, puteți configura Linux pentru a lucra integral în mediul grafic furnizat de X Window System.

Acest capitol explică modul de inserție al promptului de intrare în sistem (login) într-o fereastră grafică. De asemenea, veți învăța despre fwm2, un program de gestionare a ferestrelor frecvent folosit în Linux.



Referință încrucișată

Dacă încă nu ați instalat Linux și X, reveniți la capitolul 1 pentru a instala Linux și X. Apoi, reveniți la acest capitol pentru a testa tehnicile prezentate aici.

Lansarea automată a sistemului X la login

Acum, trebuie să intrați în sistem la un prompt de sistem și apoi trebuie să tastați `startx` pentru a lansa X Window System. Dacă întotdeauna intrați în sistem și apoi comutați imediat la ecranul grafic folosind comanda `startx`, puteți automatiza cu ușurință această operație prin adăugarea comenzii `startx` la un fișier special pe care procesorul de comenzi Linux – shellul – îl execută în mod automat la intrarea în sistem.

Procesorul de comenzi prestabilit în Linux se numește *Bash*, de la *Bourne again shell*. Când Bash se lansează, acesta citește și execută comenzile dintr-un fișier denumit `/etc/profile`. După procesarea fișierului `/etc/profile`, Bash procesează încă un fișier din directorul dumneavoastră de bază, folosind următoarea logică:

1. Dacă în directorul de bază există un fișier `.bash_profile`, Bash îl procesează.
2. Dacă Bash nu găsește fișierul `.bash_profile`, caută și apoi procesează fișierul denumit `.bash_login`, cu condiția ca acest fișier să existe.
3. Dacă directorul de bază nu conține nici unul din fișierele `.bash_profile` sau `.bash_login`, Bash caută și procesează fișierul denumit `.profile`.

Astfel, pentru a asigura lansarea automată a sistemului X, puteți insera comanda `startx` în fișierul denumit `.bash_profile` din directorul de bază. Există însă o problemă. Dacă sistemul dumneavoastră Linux se află într-o rețea, puteți intra în sistemul dumneavoastră prin rețea. Cu toate acestea, când intrați în rețea, nu este de dorit să lansați X, deoarece acesta va rula pe monitorul dumneavoastră, chiar dacă nu vă găsiți acolo.

Când executați login în calculatorul dumneavoastră Linux PC de la tastatura calculatorului (deci nu mai executați login din rețea), o variabilă de mediu denumită `TERM` este setată la șirul `linux`. Pentru a garanta că Bash execută comanda `startx` numai la efectuarea de login de la consolă (termenul se referă în mod colectiv la tastatura și la monitorul calculatorului), puteți insera comenzile adecvate pentru a determina Bash să ruleze `startx` numai când variabila `TERM` este setată la valoarea `linux`. Veți învăța mai multe despre comenzile Bash în capitolul 6, dar în continuare sunt prezentate elementele necesare din fișierul `.bash_profile` pentru a lansa X când se intră în sistem de la consolă:

```
if [ "$TERM" = "linux" ]; then
    startx
fi
```

O configurare mai interesantă o constituie utilizarea permanentă a unui mediu grafic. Chiar și promptul de intrare în sistem este o casetă de dialog. În secțiunea următoare vom explica modul de configurare a sistemului Linux pentru a furniza un ecran de login grafic.

Variabila de mediu

O variabilă de mediu este pur și simplu o modalitate de a asocia un nume cu un șir. Shellul dispune de un set standard de variabile de mediu. De exemplu, variabila de mediu `HOME` este setată la directorul dumneavoastră de bază. Modul de definire a unei variabile de mediu depinde de interpretorul de comenzi folosit. În Bash (interpretorul prestabilit pentru Linux), variabila de mediu se configurează prin utilizarea următoarei sintaxe:

`NAME=Value; export NAME`

`NAME` este numele variabilei de mediu, iar `Value` este un șir care reprezintă valoarea asociată numelui. O modalitate mai scurtă de a defini și de a folosi o variabilă export este următoarea sintaxă:

`export NAME=Value`

Configurarea unui login grafic

Pentru a intra în sistem pe numeroase tipuri de stații de lucru UNIX (cum ar fi cele produse de Hewlett-Packard sau IBM), trebuie să introduceți numele și parola într-o fereastră de dialog pentru intrarea în sistem. Configurarea stației de lucru Linux pentru a afișa un ecran grafic de intrare în sistem este simplă. De fapt, CD-ROM-ul Linux include programul X Display Manager (`xdm`), proiectat pentru a rezolva intrările în sistem prin intermediul unei interfețe grafice.

Programul X Display Manager (XDM)

Programul `xdm` este similar proceselor `getty` și `login`, care sunt frecvent folosite în sistemele UNIX (inclusiv Linux) pentru a permite unui utilizator să intre în sistem.

Când programul `xdm` rulează, acesta citește diferiți parametri de configurare dintr-un fișier denumit `xdm-config`, care de regulă se află în directorul `/etc/X11/xdm`. Unul dintre parametrii incluși în acest fișier este numele unui alt fișier, care specifică ecranele pe care `xdm` afișează

fereastra de dialog pentru intrare în sistem. Această listă se află, de obicei, în fișierul `/etc/X11/xdm/Xservers`.

Numele unui ecran mai indică și localizarea sa – dacă ecranul este local sau la distanță. Să presupunem că dorim să folosim `xdm` pentru a accesa stația dumneavoastră de lucru din ecranul local. În acest caz, `Xservers` va conține următoarea linie:

```
:0 local /usr/X11R6/bin/X
```

Specificați ecranul local prin intermediul cuvântului cheie `local`, care urmează numărului ecranului. În acest caz, trebuie să furnizați numele de cale complet al programului server X.

Procese

În UNIX, termenul *proces* se referă la un program care se execută în memorie și mediul său asociat. Acest mediu include fișierele de intrare și ieșire care aparțin programului, precum și o colecție de variabile de mediu. În esență, un proces este creat la fiecare introducere a unei comenzi la promptul interpretorului. Desigur, shellul (interpretorul de comenzi al sistemului de operare) nu este decât un alt proces, care creează procese noi la comanda utilizatorului.

Pentru configurația prestabilită (după instalarea sistemelor Linux și X de pe CD-ROM-ul anexat), `xdm` gestionează sesiunea de login după cum urmează:

1. `xdm` lansează serverul X și execută scriptul `xsetup_0` din directorul `/etc/X11/xdm`. Scriptul prestabilit `xsetup_0` rulează programul `xconsole`, care deschide o fereastră în care apar toate mesajele de la consolă.
 2. `xdm` afișează o fereastră de dialog, ce conține câmpuri unde utilizatorul poate introduce un nume și o parolă.
 3. După ce utilizatorul introduce un nume și o parolă și apasă Enter, `xdm` verifică parola și execută scriptul de inițializare `/etc/X11/xdm/GiveConsole`.
 4. `xdm` rulează scriptul `xsession` din directorul `/etc/X11/xdm`. De obicei, scriptul respectiv caută un alt fișier script, denumit `.xsession`, în directorul de bază al utilizatorului. Acest fișier este similar cu fișierul `.xinitrc` folosit de `xinit` după lansarea serverului X. Fișierul `.xsession` trebuie să conțină comenzile pentru lansarea aplicațiilor X selectate. De regulă, această listă include emulatorul de terminal `xterm`. Dacă nu există nici un fișier `.xsession`, scriptul `xsession` rulează scriptul `/etc/X11/xinit/xinitrc`.
 5. Din acest moment, utilizatorul poate interacționa cu sistemul prin intermediul ecranului X și al programului de gestionare a ferestrelor (lansat de fișierul `.xsession` sau de script-ul de sistem `/etc/X11/xinit/xinitrc`).
 6. Când utilizatorul părăsește X (prin închiderea programului de gestionare a ferestrelor), `xdm` rulează scriptul `/etc/X11/xdm/TakeConsole`. La finalul acestui script, `xdm` revine la fereastra de dialog de intrare în sistem și așteaptă intrarea unui alt utilizator în sistem.
- Deși această succintă descriere a utilitatii `xdm` nu prezintă toate detaliile, `xdm` furnizează o interfață prietenoasă pentru intrarea în sistem. De asemenea, ca și multe alte programe utilitare X, `xdm` este extrem de configurabil. De exemplu, numele fișierelor script din lista anterioară sunt numele obișnuite, dar utilizatorul poate specifica alte nume prin intermediul fișierului denumit `xdm-config`, care, în mod prestabilit, se află de asemenea în directorul `/etc/X11/xdm`.
- Când instalați Linux și X de pe CD-ROM-ul anexat acestei cărți, `xdm` este deja configurat pentru a afișa o fereastră de dialog pentru intrarea în sistem pe monitorul calculatorului

dumneavoastră. Dacă doriți să testați programul `xdm`, puteți intra ca root și tasta comanda `xdm -nodaemon` la promptul interpretorului de comenzi. Această comandă determină lansarea programului `xdm`, care la rândul său, afișează fereastra de dialog pentru intrarea în sistem. În acest moment, dumneavoastră sau un alt utilizator puteți intra în sistem prin introducerea numelui utilizatorului și a parolei în fereastra de dialog. Dar încă nu dispuneți de un aranjament în care `xdm` să ruleze în mod automat imediat după încărcarea sistemului de operare Linux. Puteți crea un astfel de aranjament destul de simplu, dar mai întâi trebuie să înțelegeți care este modul de rulare a programelor după încărcarea sistemului de operare.

Procesul init

Așa cum ați aflat din etapele de instalare din capitolul 1, când este încărcat sistemul de operare Linux, acesta încarcă și rulează programul sistem de operare fundamental (core) de pe disc. Sistemul de operare fundamental este proiectat însă pentru rularea altor programe. Un proces denumit `init` este responsabil pentru lansarea în execuție a setului inițial de procese de pe sistemul dumneavoastră Linux.

Ce anume lansează procesul `init` depinde de următoarele:

- **Nivelul de rulare**, care desemnează o configurație de sistem în cadrul căreia există numai un grup selectat de procese
- Fișierul `/etc/inittab`, un fișier text care specifică procesele de lansat la diferite niveluri de rulare



Secret

Nivelul de rulare curent, alături de conținutul fișierului `/etc/inittab`, controlează procesele lansate de `init`. Linux, de exemplu, are șapte niveluri de rulare, numerotate de la 0 la 6. Prin convenție, unele dintre aceste niveluri indică anumite procese care rulează la nivelul respectiv. De exemplu, nivelul de rulare 1 desemnează un sistem independent, cu un singur utilizator. Nivelul de rulare 0 semnifică un sistem oprit, iar nivelul 6 arată că se execută `reboot`. Nivelurile de rulare 2-5 sunt moduri multiple utilizator, cu diferite niveluri de capacitate.

Nivelul prestabilit de rulare inițial este 3. Așa cum se va explica în secțiunea următoare, puteți modifica nivelul de rulare prestabilit prin editarea unei linii din fișierul `/etc/inittab`.

Fișierul /etc/inittab

Fișierul `/etc/inittab` reprezintă cheia spre înțelegerea proceselor lansate de `init` la diferite niveluri de rulare. Puteți examina conținutul fișierului (pentru aceasta nu trebuie să intrați în sistem ca root) folosind următoarea comandă:

```
more /etc/inittab
```

În continuare sunt prezentate câteva linii selectate din fișierul prestabilit `/etc/inittab`:

```
#
# inittab      This file describes how the INIT process should
#             set up the system in a certain run-level.
#
# Author:      Miquel van Smoorenburg.
#              <miquels@drinkel.nl.mugnet.org>
#              Modified for RHS Linux by Mark Ewing and Donnie Barnes
#
#
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
```

```
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left. Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have power installed and your UPS
# connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr::12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

Linii care încep cu simbolul diez (#) sunt *comentarii*. Prima linie *non-comentariu* din fișierul /etc/inittab specifică nivelul prestabilit, după cum urmează:

```
id:3:initdefault:
```

Chiar dacă nu cunoașteți sintaxa fișierului /etc/inittab (și nici nu trebuie să o învățați), probabil vă puteți da seama că cifra 3 din linia respectivă reprezintă nivelul de rulare prestabilit. Astfel, dacă doriți ca sistemul dumneavoastră să fie la nivelul de rulare 5 după inițializare, nu trebuie decât să înlocuiți pe 3 cu 5.



Sugestie

Tastați comanda **man inittab** pentru a vedea sintaxa detaliată a intrărilor din fișierul inittab. În continuarea acestei secțiuni este descris formatul de fișier inittab.

Fiecare intrare din fișierul /etc/inittab specifică un proces pe care init trebuie să-l inițializeze la unul sau mai multe niveluri de rulare specificate – pur și simplu listați toate nivelurile de rulare la care trebuie să ruleze procesul. Fiecare intrare din fișierul inittab are patru câmpuri – separate prin două puncte – în următorul format:

```
id:niveluri_de_rulare:actiune:proces
```

■ Câmpul *id* este un identificator unic, format din unul sau două caractere. Procesul init folosește acest câmp în mod intern. Puteți utiliza orice identificator doriți, dacă nu folosiți același identificator pe mai multe linii.

■ Câmpul *niveluri_de_rulare* este o secvență din zero sau mai multe caractere, unde fiecare caracter semnifică un nivel de rulare. De exemplu, linia cu identificatorul 1 se aplică nivelurilor de rulare cuprinse între 1 și 5, deci valoarea câmpului *niveluri_de_rulare* pentru această intrare este 12345.

■ Câmpul *actiune* indică procesului init cum să procedeze cu intrarea respectivă. De exemplu, dacă acest câmp este initdefault, init interpretează câmpul *niveluri_de_rulare* ca nivel de rulare prestabilit. Dacă acest câmp este setat la wait, init lansează procesul specificat în câmpul *proces* și așteaptă încheierea procesului.

■ Câmpul *proces* specifică procesul pe care init trebuie să îl lanseze. Desigur, unele setări ale câmpului *actiune* nu necesită nici un câmp *proces*. (Când *actiune* este initdefault, de exemplu, nu este nevoie de nici un câmp *proces*.)

Procesul este deseori desemnat sub forma unui shell script (script pentru interpretorul de comenzi), care, la rândul său, poate lansa mai multe procese. De exemplu, intrarea 13 este specificată sub forma:

```
13:3:wait:/etc/rc.d/rc 3
```

Această intrare arată că init trebuie să execute fișierul /etc/rc.d/rc cu 3 ca argument. Dacă examinați fișierul /etc/rc.d/rc, veți observa că este un fișier script. După ce veți învăța mai multe despre scrierea scripturilor de fișier în capitolul 6, puteți studia fișierul /etc/rc.d/rc pentru a vedea modul de pornire pentru nivelurile de rulare 1-5.

Revenind la subiectul lansării programului xdm pentru un login cu interfață grafică, puteți vedea o referință explicită la xdm în ultima linie a fișierului /etc/inittab, care este definit după cum urmează:

```
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

Această comandă determină rularea programului xdm. Datorită unei legături simbolice, /usr/X11/bin/xdm și /usr/X11R6/bin/xdm sunt referința la același program.

Pentru a lansa xdm în mod automat după încărcarea sistemului de operare, nu trebuie decât să editați fișierul /etc/inittab pentru a înlocui cifra 3 din linia următoare cu 5 (modificând astfel nivelul de rulare prestabilit la 5):

```
id:3:initdefault:
```



Atenție!

Înainte de a edita fișierul /etc/inittab, trebuie să știți că orice eroare apărută în acest fișier poate preveni încărcarea sistemului Linux până la punctul în care se poate executa login. Dacă nu puteți intra în sistem, nu vă puteți folosi sistemul. Așa cum se va arăta în secțiunea următoare, trebuie să încercați nivelul de rulare 5 cu comanda **init 5** înainte de a modifica efectiv nivelul de rulare prestabilit în cadrul fișierului /etc/inittab.

Comanda init

Pentru a testa un nou nivel de rulare, nu sunteți obligat să modificați nivelul de rulare prestabilit din fișierul `/etc/inittab`. Dacă intrați în sistem ca `root`, puteți modifica nivelul de rulare (și implicit setul de procese care rulează în Linux) folosind comanda `init`, care are următorul format:

```
init runlevel
```

În acest caz, `runlevel` trebuie să fie un singur caracter, reprezentând nivelul de rulare dorit. De exemplu, pentru a pune sistemul în mod utilizator unic, se introduce comanda:

```
init 1
```

Astfel, dacă trebuie să testați nivelul de rulare 5 (și să vedeți dacă `xdm` funcționează) fără a modifica fișierul `/etc/inittab`, introduceți următoarea comandă la promptul procesorului de comenzi:

```
init 5
```

Sistemul își va încheia toate procesele curente și va încărca nivelul de rulare 5. În mod prestabilit, comanda `init` așteaptă 20 de secunde înainte de a opri toate procesele curente și de a lansa noile procese pentru nivelul de rulare 5.

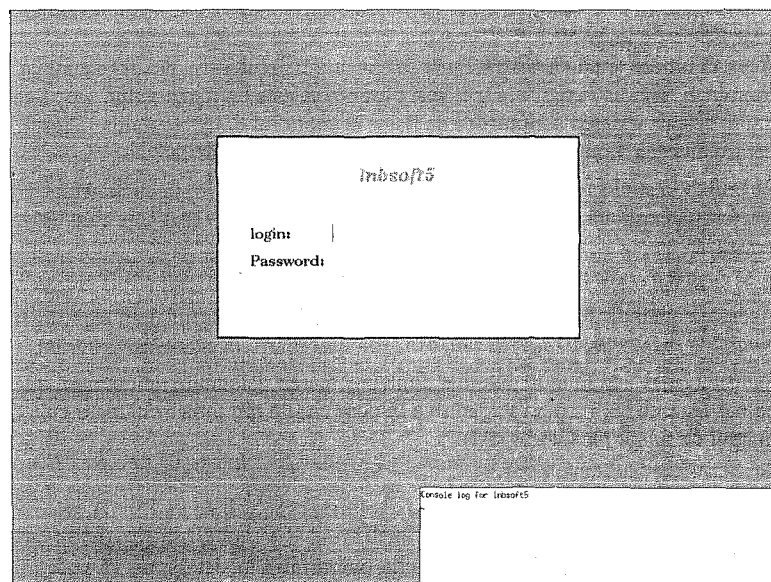


Pentru a comuta imediat la nivelul de rulare 5, tastați comanda `init -t0 5`. (Numărul plasat după opțiunea `-t` indică numărul de secunde care trebuie să treacă pentru ca `init` să modifice nivelul de rulare.)

Procesul `init` comută la nivelul de rulare 5 și lansează programul `xdm`. Programul `xdm`, la rândul său, lansează `X` și afișează o casetă de dialog pentru intrare în sistem, așa cum se vede în figura 5.1.

Figura 5-1

Programul `xdm` afișează o fereastră grafică de login.



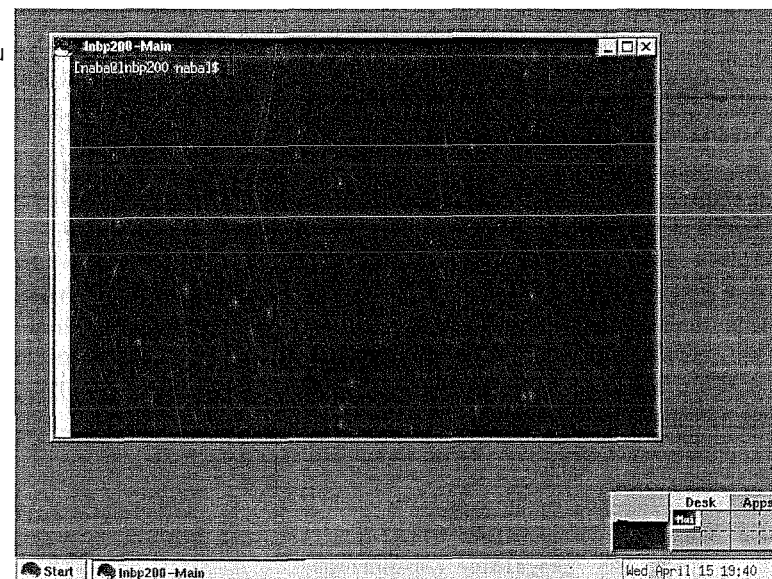
Cursorul pentru introducerea textului (similar unei bare verticale) apare în câmpul `login:`. Tastați numele dumneavoastră de login și apăsați `Enter`. Cursorul text se deplasează spre câmpul `Password:`. Introduceți parola și apăsați `Enter` pentru a finaliza procesul de intrare în sistem.

După ce ați intrat în sistem de la fereastra login `xdm`, `X` se lansează ca de obicei și rulează aplicațiile specificate în fișierul `xclients`, localizat în directorul `/etc/X11/xinit`. Dacă doriți să vă lansați propriul dumneavoastră set de aplicații, folosiți un editor de texte pentru a pregăti un fișier denumit `.xclients` în directorul dumneavoastră de bază și includeți în fișierul respectiv comenzi pentru lansarea diferitelor aplicații, precum `xterm` și programul de gestionare a ferestrelor `fwm2`.

Suprafața de lucru (desktop) inițială generată de comenzile incluse în fișierul `/etc/X11/xinit/xclients` include o fereastră terminal `xterm` și programul de gestionare a ferestrelor `fwm2`, cu o bară de stare și câteva butoane, așa cum se vede în figura 5.2.

Figura 5-2

Suprafața de lucru inițială a sistemului `X`, cu fișierele de configurare prestabilite.



Personalizarea sistemului X

Puteți personaliza aproape toate aspectele interfeței grafice a sistemului `X`, de la culoarea de fundal a ecranului la fontul prestabilit pentru toate aplicațiile. Filozofia de bază a sistemului `X` (ca și a sistemului `UNIX`, de altfel) este ca utilizatorii să aibă libertatea de configura programele conform propriilor preferințe. Astfel, aplicațiile `X` pot încărca diverse setări de la un fișier sursă. De asemenea, fiecare aplicație `X` acceptă un set de opțiuni în linie de comandă standard. Această secțiune vă prezintă metode generale de personalizare a aplicațiilor `X`, dar și informații privind emulatorul de terminal `xterm` și programul de gestionare a ferestrelor `fwm2`.

Aspectul ferestrei root

În `X`, întregul ecran se numește fereastră `root` (sau fereastră rădăcină). Un program utilitar denumit `xsetroot` vă permite să modificați aspectul ferestrei `root`. Prin intermediul acestui program, puteți personaliza ecranul după cum urmează:

- Opțiunea `-solid` servește la stabilirea pentru ecran a unei culori uniforme. Puteți specifica culoarea prin numele acesteia; numele culorilor se găsesc în fișierul `/usr/lib/X11/rgb.txt`.

- Opțiunea -bitmap se folosește pentru setarea fundalului la un fișier grafic de tip bitmap.
- Opțiunile -cursor sau -cursor_name se folosesc pentru a specifica o formă de cursor atunci când mouse-ul se află amplasat într-o anumită regiune a ferestrei root.
- Opțiunea -gray se folosește pentru setarea fundalului la culoarea gri.

Pentru a personaliza fereastra root, trebuie să rulați xsetroot de la prompt-ul de comandă, după care puteți testa diferitele opțiuni până când obțineți un aspect care vă satisface. Apoi, puteți insera comenzile xsetroot în fișierul dumneavoastră .Xclients.

Dacă aveți experiență în lucrul cu mediile grafice Macintosh sau Windows 95, puteți observa că operațiile executate de comanda xsetroot sunt similare cu personalizarea suprafeței de lucru din Control Panel. Desigur, în Macintosh și Windows 95 se folosește un GUI pentru personalizarea suprafeței de lucru. În X, va trebui să vă acomodați cu un program în linie de comandă (xsetroot).

Culoarea ecranului

Puteți stabili culoarea de fond a ecranului folosind programul xsetroot. De exemplu, pentru a seta culoarea de fond la o culoare uniformă albastru-oțel (SteelBlue), introduceți următoarea comandă la promptul interpretorului de comenzi dintr-o fereastră xterm:

```
xsetroot -solid SteelBlue
```

Pentru a încerca mai multe culori, trebuie să cunoașteți numele acestora. Definițiile culorilor se află într-un fișier text denumit /usr/lib/X11/rgb.txt. Pentru a vedea o listă a numelor de culori, deschideți fișierul respectiv cu un editor de texte sau tastați următoarea comandă într-o fereastră xterm:

```
more /usr/lib/X11/rgb.txt
```

În continuare sunt prezentate câteva rânduri din fișierul rgb.txt:

```
255 255 255 white (alb)
0 0 0 black (negru)
190 190 190 gray (gri)
65 105 225 RoyalBlue (albastru regal)
70 130 180 SteelBlue (albastru oțel)
218 165 32 goldenrod
184 134 11 DarkGoldenrod
233 150 122 DarkSalmon
255 160 122 LightSalmon
```

Așa cum vă puteți da seama, cele trei numere indică nivelurile de intensitate ale componentelor de roșu, verde și albastru ale culorii. Nivelurile de intensitate variază între 0 și 255.



În afară de utilizarea numelui descriptiv al culorii, puteți specifica o culoare sub forma unui număr hexazecimal, format din șase cifre hexazecimale, care folosește maximum două cifre pentru fiecare dintre cele trei componente ale unei culori: roșu, verde și albastru. Cifrele dintr-un număr hexazecimal cu șase cifre sunt interpretate de la stânga la dreapta, cu cea mai semnificativă pereche de cifre pentru roșu, a doua pereche pentru verde și a treia pentru albastru. Caracterul # precede cifrele hexazecimale, pentru a simboliza o specificație de culoare în format hexazecimal. Când specificați o culoare în acest mod, trebuie să inserați valoarea hexazecimală a culorii între semnele citării. Puteți folosi formatul hexazecimal pentru a specifica orice culoare doriți.

Tabelul 5.1 prezintă unele culori frecvent folosite, exprimate în format hexazecimal:

Tabelul 5.1 Culori comune în hexazecimal

Culoare	Valoare în hexazecimal
negru	#000000
roșu	#ff0000
verde	#00ff00
albastru	#0000ff
galben	#ffff00
cyan	#00ffff
magenta	#ff00ff
alb	#ffffff

Acum că știți să specificați orice culoare arbitrară în format hexazecimal RGB, puteți testa culori noi. De exemplu, puteți încerca o culoare de fond mai ciudată, folosind următoarea comandă:

```
xsetroot -solid "#ccb0b0"
```

Dacă doriți un fundal de culoare gri (care poate fi foarte plăcută pentru ochi, deși puțin plictisitoare), puteți folosi comanda:

```
xsetroot -gray
```

Numere hexazecimale

Chiar dacă nu cunoașteți terminologia, sunteți la curent cu un sistem de numerație din experiența dumneavoastră zilnică, mai exact sistemul de numerație în baza 10. Numerele în baza 10 se scriu folosind cifrele de la 0 la 9, iar când scriem valori ne bazăm pe conceptele de unități, zeci, sute, mii etc. De exemplu, să considerăm numărul 3495. Folosind conceptele de unități, zeci, sute și mii, putem scrie următoarele:

3495 = 3 mii + 4 sute + 9 zeci + 5 unități

sau

$3495 = 3 \times 10^3 + 4 \times 10^2 + 9 \times 10^1 + 5 \times 10^0$

Ultima formă reprezintă numărul scris folosind puteri ale lui 10, care reprezintă baza sau rădăcina sistemului zecimal de numerație.

Sistemul de numerație hexazecimal este similar cu sistemul zecimal de numerație, cu deosebirea că baza este 16. Pentru a scrie un număr în hexazecimal, aveți nevoie de 16 cifre care să reprezinte valorile de la 0 la 15, așa cum sistemul zecimal folosește cifrele de la 0 la 9. Iată cele 16 cifre din sistemul hexazecimal:

Cifra zecimală	Valoarea cifrei în hexazecimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
a	10
b	11
c	12
d	13
e	14
f	15

După cum puteți vedea, literele de la a la f, sunt folosite pentru a reprezenta valorile cuprinse între 10 și 15 inclusiv. Pentru cifrele hexazecimale se pot folosi majuscule sau litere mici.

Dacă cunoașteți asemănarea între sistemele de numerație zecimal și hexazecimal, puteți descoperi cu ușurință echivalentul în zecimal al unui număr în hexazecimal. De exemplu, 1b în hexazecimal este $1 \times 16^1 + 11 \times 16^0 = 27$ în zecimal.

Imaginea de fond a ecranului

Nu este necesar să optați pentru o culoare uniformă ca fundal pentru ecran. Puteți folosi comanda `xsetroot` cu opțiunea `-bitmap` pentru a placa fundalul ecranului cu o imagine bitmap monocromă (placare înseamnă așezarea una lângă alta a mai multor copii ale imaginii până la acoperirea întregului ecran – similar cu acoperirea podelei cu plăci de gresie). Puteți folosi opțiunile `-fg` și `-bg` pentru a trasa imaginea în culori de fundal și pentru a umple fundalul cu o culoare de fond selectată.

Încercați următoarea comandă pentru a seta fundalul ecranului la un model interesant al emblemei sistemului X:

```
xsetroot -bitmap /usr/include/X11/bitmaps/xlogo64 -fg "#cccccc" -bg "#c0c0c0"
```

Directorul `/usr/include/X11/bitmaps` conține cu mult mai multe fișiere bitmap, pe care le puteți încerca drept imagini de fond.

Deși `xsetroot` nu poate afișa o imagine color ca fundal de ecran, puteți folosi programul `xv` pentru a afișa ca fundal un fișier imagine ales de dumneavoastră. Programul `xv` poate manipula o mare varietate de formate de imagine, între care și formatele frecvent folosite GIF, PCX, BMP, JPEG și TIFF. Să presupunem că aveți un fișier imagine preferat, denumit `sunset.gif`, care prezintă un superb apus de soare, iar dumneavoastră doriți să folosiți această imagine ca fundal pe ecranul X. Pentru a executa această operație, includeți următoarea comandă în fișierul dumneavoastră `.xsession`:

```
xv -quit -root -max /usr/local/sunset.gif
```

Această comandă pornește de la premisa că fișierul imagine `sunset.gif` se află în directorul `/usr/local`.



Sugestie

Evitați utilizarea de imagini colorate ca fundal de ecran; dacă afișați o imagine de fond cu prea multe culori, celelalte programe X vor rămâne fără culori. Nu uitați că o placă video normală pe 8 biți poate afișa maximum 256 de culori. Dacă imaginea de fond are mai mult de 16 culori, puteți avea probleme la rularea altor programe. În particular, programe precum Netscape Navigator (browser de Web) au nevoie de culori pentru a afișa imaginile incluse în majoritatea paginilor Web.

În afară de a determina epuizarea rezervei de culori a sistemului X, un fundal colorat impune o utilizare a resurselor serverului X și poate afecta performanțele sistemului. Ca atare, o soluție sigură o constituie utilizarea unei culori uniforme pentru fundalul ecranului.

Forma și culoarea cursorului

Prin intermediul programului `xsetroot`, puteți determina forma și culoarea cursorului atunci când indicatorul de mouse se găsește deasupra unei suprafețe oarecare din fereastra root. În interiorul ferestrelor unei aplicații, aplicația este cea care controlează forma și culoarea cursorului.

Deși puteți folosi două fișiere bitmap pentru a defini un cursor de formă aleasă arbitrar, o metodă mai simplă rezidă în selecția uneia dintre formele de cursor predefinite. Puteți folosi opțiunea `-cursor_name` a programului `xsetroot` pentru a specifica un cursor. Numele de cursor sunt definite în fișierul `/usr/include/X11/cursorfont.h`. Numele sunt destinate a fi folosite în programele X, iar fiecare cursor are un prefix `xc_`. Acest prefix trebuie omis când se folosește numele cursorului în programul `xsetroot`.

În continuare sunt date câteva nume de cursor:

<code>x_cursor</code>	Un cursor în formă de X (cursorul prestabilit)
-----------------------	--

<code>arrow</code>	O săgeată orientată spre colțul din dreapta-sus
<code>left_ptr</code>	O săgeată orientată spre colțul din stânga-sus

Cursorul prestabilit este un cursor în formă de X, de culoare neagră. Pentru a transforma cursorul într-o săgeată albă orientată spre stânga, tastați următoarea comandă:

```
xsetroot -cursor_name left_ptr -fg white -bg black
```

În mediile grafice Macintosh și Windows 95, cursorul este o săgeată orientată spre stânga. Dacă cunoașteți aceste medii, poate doriți să transformați cursorul într-o săgeată orientată spre stânga.

Resurse X

Majoritatea aplicațiilor X sunt extrem de configurabile. Puteți modifica aspectul și, într-o oarecare măsură, chiar și comportamentul unei aplicații X, în două moduri:

- Modificați comportamentul prestabilit al aplicației prin intermediul unor opțiuni specificate în linia de comandă care determină lansarea aplicației.
- Specificați valori pentru opțiuni într-un fișier text, denumit *fișier resursă*. (Programatorii X folosesc termenul *resursă* pentru a desemna orice opțiune a unei aplicații configurabilă de către utilizator.)

Pentru utilitare UNIX mici, câteva opțiuni în linie de comandă pot fi suficiente pentru a specifica toți parametrii configurabili de către utilizator ai programului. Aplicațiile X au tendința de a avea un mare număr de variabile modificabile de către utilizator. O aplicație X poate avea numeroase ferestre, iar fiecare fereastră poate avea o margine, o culoare de fundal și o culoare de prim-plan. Deși o aplicație X furnizează valori prestabilite pentru acești parametri, utilizatorul poate șterge majoritatea acestor valori. De asemenea, utilizatorul poate modifica fontul oricărui text care urmează a se afișa în ferestre. Cu alte cuvinte, aplicațiile X au prea multe variabile pentru a permite utilizatorului configurarea acestora folosind opțiunile din linia de comandă. Creatorii sistemului X au recunoscut această problemă și au conceput o bază de date cu resurse.

În cadrul unei aplicații X, termenul *resursă* se referă la orice parametru care afectează comportamentul și modul de prezentare ale unei aplicații. În consecință, culorile de fond și de prim-plan, fonturile și dimensiunea și amplasamentul ferestrelor sunt exemple de resurse. O resursă nu trebuie să fie un parametru legat de sistemul X, ci poate fi specificat de utilizator. O aplicație poate avea un parametru denumit *verbose*, care permite (când este activat) tipărirea de informații detaliate. În afară de dimensiunea și locația unei ferestre de aplicație, *verbose* poate fi considerat ca resursă a aplicației.

Fișierul resursă

Resursele unei aplicații X trebuie specificate într-un fișier text, cunoscut sub numele de *fișier resursă* sau *bază de date cu resurse*. Baza de date cu resurse X este un simplu fișier text în care puteți specifica valorile mai multor parametri într-un format bine definit.

Fișierele resursă X nu sunt atât de complicate și sofisticate ca o bază de date tradițională. Baza de date cu resurse X conține specificații de forma „toate culorile de fond sunt albe“, „fundalul programului `xterm` este cyan deschis“ etc.

Aplicațiile X folosesc un set de rutine utilitare, cunoscute sub numele colectiv de *program de gestionare a resurselor X* – pentru a extrage valoarea parametrilor individuali identificați cu exactitate din această bază de date oarecum imprecisă. Să considerăm interogarea „care este culoarea de prim-plan a aplicației `xterm`“? Dacă ați specificat culoarea de prim-plan a aplicației

xterm în fișierul resursă, programul de gestionare a resurselor returnează valoarea solicitată. Dacă unica specificație pentru toate culorile de prim-plan din baza de date este instrucțiunea generală „toate culorile de prim-plan sunt negre“, atunci valoarea returnată pentru culoarea de prim-plan a aplicației xterm este de asemenea neagră.

Convenția de nume a resurselor X

Pentru a specifica valorile resurselor dintr-o aplicație X, cum ar fi culoarea de fond, trebuie să știți cum se denumește o resursă și cum se specifică valoarea unei resurse.

Numele unei resurse depinde de numele aplicației și de numele componentelor acesteia, care de regulă sunt principalele ferestre copil. Pentru aplicații construite folosind *trusa de instrumente X* (*X toolkit*) – care reprezintă, în esență, o bibliotecă de componente ale interfeței cu utilizatorul – componentele vor fi numele primitivelor folosite pentru generarea aplicației.



Observație

O *primitivă (widget)* nu este nimic mai mult decât o componentă a interfeței cu utilizatorul, precum un buton, o casetă listă sau o casetă de dialog. O *trusă de instrumente X*, cum este trusa Motif, furnizează funcțiile utilizabile de programatori pentru crearea și utilizarea de primitive în programele acestora.

Numele aplicației și ale componentelor acesteia pot fi de două tipuri: *numele clasei și numele instanței*. Numele clasei simbolizează categoria generală din care face parte aplicația sau componenta, în timp ce fiecare copie poate avea propriul său nume de instanță.

Să considerăm un exemplu concret: aplicația xterm. Această aplicație aparține clasei XTerm, iar instanța are numele xterm. Aplicația xterm folosește o componentă denumită vt100 de clasă VT100 (componenta vt100 emulează un VT102, dar numele intern este VT100), care conține o componentă denumită scrollbar (bară de derulare) din clasa Scrollbar. Să considerăm acum următoarele resurse: culoarea de fond a ferestrei VT100 și vizibilitatea barei de derulare în fereastra respectivă. În xterm, ca în majoritatea aplicațiilor X, resursa culoare de prim-plan are numele Foreground, iar numele instanței este foreground. Vizibilitatea barei de derulare este controlată de o variabilă logică denumită on.

Majoritatea aplicațiilor X respectă convenția de denumire a clasei unei resurse prin scrierea cu majusculă a numelui instanței. Numele de aplicație respectă de asemenea această convenție, care asigură o consecvență între aplicații. Numele aplicației, componentele acesteia și resursele, cu toate acestea, se află sub controlul integral al aplicației. (Cu toate acestea, aplicațiile create cu truse de instrumente sunt oarecum limitate de numele încorporate ale primitivelor predefinite.)

Acum puteți atribui acestor resurse nume unice în xterm și puteți specifica valori pentru acestea. Puteți atribui valori pentru culoarea de prim-plan a ferestrei VT100 și pentru variabila on a barei de derulare după cum urmează:

```
xterm.vt100.foreground: yellow
xterm.vt100.scrollbar.on: true
```

Acest exemplu prezintă sintaxa folosită la denumirea resurselor și la atribuirea de valori pentru acestea. Numele unei resurse începe cu numele aplicației, urmat de numele componentelor, separate printr-un punct (.). Numele resursei este ultimul, iar valoarea resursei este precedată de două puncte (:). Valoarea este specificată printr-un șir text, a cărui interpretare este la latitudinea aplicației. Programul de gestionare a resurselor dispune de rutine utilitare care asistă programatorii la această operație.

Numele prezentate până acum în această secțiune reprezintă nume de instanță complete, care conțin atât numele aplicației, cât și pe al tuturor componentelor acesteia. De asemenea, puteți

avea nume de clasă complete, care pentru xterm.vt100. foreground este XTerm.VT100.Foreground. Veți obține acest nume prin înlocuirea numelor de instanță ale fiecărei componente cu numele clasei corespunzătoare.

Nume parțiale ale resurselor

În secțiunea anterioară am arătat modul de specificare a unei resurse prin numele său complet, dar specificația resursei poate fi imprecisă. Puteți specifica, de exemplu, că toate componentele clasei VT100 trebuie să aibă un prim-plan de culoare galbenă. Puteți realiza această operație prin specificarea următoarei intrări în baza de date cu resurse:

```
*VT100.Foreground: yellow
```

Deoarece nu s-a specificat un nume de aplicație, această precizare a culorii de prim-plan se aplică componentei VT100 folosite în orice aplicație. Similar, pentru a specifica culoarea de fond a fiecărei componente a aplicației xterm ca fiind Navy, veți include următoarea linie în fișierul resursă:

```
xterm*background:Navy
```

Pentru a înțelege această schemă de denumire a resurselor, trebuie să cunoașteți anumite aspecte privind mecanismele interne ale programului de gestionare a resurselor X. Până acum, ați intuit probabil că programul de gestionare a resurselor X localizează valoarea unei resurse prin echivalența dintre un nume de resursă bine precizat și intrările imprecise din baza de date cu resurse. Algoritmul de căutare folosit de programul de gestionare a resurselor respectă anumite reguli pentru stabilirea unei echivalențe între numele complet al resursei și numele parțiale din baza de date cu resurse. Cunoașterea acestor reguli vă ajută să înțelegeți categoria de precizare pentru o resursă care este suficient de exactă pentru satisfacerea anumitor cerințe:

■ Un asterisc (*) simbolizează zero sau mai multe componente ale numelui. Ca atare, interogarea pentru xterm.vt100.foreground satisface următoarea intrare:

```
xterm*foreground: yellow
```

■ După explicarea asteriscului, numele aplicației, numele componentelor și numele resursei (clasă sau instanță) trebuie să fie echivalente cu articolele prezente în intrare. Ca atare, o interogare pentru XTerm.VT100.Scrollbar.Background satisface următoarea intrare:

```
xterm.vt100.scrollbar.background: Navy
```

dar nu și pe următoarea

```
xterm.vt100.scrollbar.on: true
```

■ Specificațiile de resursă mai precise au prioritate față de cele mai puțin precise. Intrările cu punct (.) au prioritate față de cele cu asterisc (*). Dacă specificați liniile de mai jos, toate elementele din xterm au un fundal de culoare navy (ultramarin), dar bara de derulare va avea un fundal alb:

```
xterm*background: Navy
```

```
xterm.vt100.scrollbar.background: white
```

■ Numele de instanță au prioritate față de numele de clasă. Astfel, intrarea specifică xterm*background va avea prioritate asupra celei care folosește nume de clasă: XTerm*Background.

■ O intrare cu un nume de clasă sau un nume de instanță are prioritate asupra unei intrări care nu folosește nici unul dintre cele două tipuri de nume. În xterm, valoarea dată de intrarea XTerm*Foreground are prioritate față de intrarea cu caracter mai general *Foreground.

■ Echivalența numelor se stabilește de la stânga la dreapta, deoarece ierarhia componentelor unui nume de resursă are sensul de la stânga la dreapta. Cu alte cuvinte, când căutați resursa denumită `xterm.vt100.scrollbar.background`, programul de gestionare a resurselor stabilește echivalența cu intrarea

```
xterm.vt100*background: white
```

în loc de

```
xterm*scrollbar.background: Navy
```

deoarece `vt100` apare la stânga numelui `scrollbar`.

Locația fișierelor resursă

Majoritatea aplicațiilor X încarcă setări de resursă din mai multe surse, într-o anumită ordine. În primul rând, aplicațiile caută un fișier denumit `/usr/lib/X11/app-defaults/AppClass`, unde `AppClass` este numele de clasă al aplicației. Astfel, `xterm` își caută resursele în fișierul `/usr/lib/X11/app-defaults/XTerm`.

În continuare, aplicația caută un șir denumit `RESOURCE_MANAGER`, care este asociat ferestrei `root` a ecranului în care apare fereastra aplicație. În X, un șir asociat cu fereastra `root` este cunoscut sub numele de *proprietate* a ferestrei `root`.



Sugestie

Puteți folosi programul utilitar `xprop` pentru a vedea dacă această proprietate există în fereastra `root` a ecranului dumneavoastră. Pur și simplu tastați `xprop -root` și căutați o intrare unde scrie `RESOURCE_MANAGER (STRING)` la ieșire. De asemenea, utilitarul `xprop` mai poate citi un fișier resursă și poate încărca conținutul acestuia în această proprietate (sub forma unui șir lung).

Dacă proprietatea `RESOURCE_MANAGER` nu există, aplicația citește specificațiile de resursă din fișierul `.xdefaults` existent în directorul dumneavoastră de bază. În continuare, aplicația încarcă resursele (dacă acestea există) specificate de fișierul indicat prin variabila de mediu `XENVIRONMENT`. Dacă această variabilă nu este setată, următoarea sursă de resurse este un fișier denumit `.xdefaults-hostname` din directorul dumneavoastră de bază. În acest fișier, `hostname` este numele sistemului pe care rulează aplicația.

Evident, puteți ignora oricare dintre specificațiile de resursă, prin intermediul opțiunilor din linia de comandă.

Dacă doriți să vedeți care este efectul resurselor, în continuare este prezentat un exemplu. În fereastra curentă a programului `xterm`, folosiți un editor de texte pentru a crea fișierul `xttest`, care conține următoarele linii:

```
xterm.vt100.scrollbar: true
*VT100.Foreground: yellow
xterm.vt100.background: navy
xterm.vt100.scrollbar.background: white
```

Acum introduceți următoarea comandă, rulați programul utilitar `xrdb` și încărcați aceste setări în proprietatea `RESOURCE_MANAGER` a ferestrei `root`:

```
xrdb -load xttest
```

Tastați `xprop -root` pentru a verifica dacă resursele au fost încărcate. ● Observați că conținutul fișierului resursă sunt stocate intern în proprietatea `RESOURCE_MANAGER` sub forma unui șir lung. Acum lansați o altă sesiune `xterm`, cu comanda `xterm &`. Veți vedea o nouă fereastră `xterm`, care folosește caractere galbene pe un fundal ultramarin, cu excepția barei de derulare, care are un fundal alb.

Resurse X standard

Prin convenție, toate aplicațiile X dispun de un set standard de resurse. Între aceste resurse amintim parametri precum culorile de fond și de prim-plan, dimensiunea și amplasamentul ferestrelor (cunoscute sub termenul colectiv de *geometrie*) și fontul. Tabelul 5.2 prezintă unele resurse X standard.

Tabelul 5.2 Resurse X standard

Nume instanță	Nume clasă	Opțiune în linia de comandă	Specifică
background	Background	-bg -background	Culoarea de fundal
borderColor	BorderColor	-bd -border	Culoarea marginii
borderWidth	BorderWidth	-bw -borderwidth	Lățimea marginii (în pixeli)
display	Display	-d -display	Numele ecranului
foreground	Foreground	-fg -foreground	Culoarea de prim-plan
font	Font	-fn -font	Numele fontului
geometry	Geometry	-geometry	Dimensiune și locație
title	Title	-title	Șirul titlu

Opțiuni în linia de comandă pentru aplicațiile X

În afară de bazele de date cu resurse, aplicațiile X acceptă și argumente în linia de comandă. Tabelul 5.2 prezintă opțiunile în linie de comandă frecvent folosite pentru aplicațiile X. În această secțiune vom prezenta pe scurt unele opțiuni importante.

Opțiunea -display

Această opțiune se folosește pentru specificarea ecranului pe care trebuie să apară datele de ieșire ale aplicației. (● aplicație X poate rula pe un sistem și își poate afișa datele de ieșire pe un ecran conectat la un alt sistem din rețea.) De exemplu, dacă lucrați la un calculator aflat la distanță și doriți să rulați clientul `xclock` pe sistemul respectiv, puteți lansa aplicația cu o comandă de genul: `xclock -display sysname: 0 &`

În acest caz, `sysname` este numele sistemului (numele pe care îl vedeți când tastați comanda `hostname` pe sistemul dumneavoastră).

Opțiunea -geometry

O altă opțiune comună este `-geometry`, folosită pentru a specifica dimensiunea și locația unei ferestre de aplicație. Geometria trebuie specificată într-un format standard, după cum urmează: `widthxheight[+-]xoffset[+-]yoffset`

În acest format, `width`, `height`, `xoffset` și `yoffset` sunt numere, iar dumneavoastră trebuie să selectați unul dintre cele două semne incluse între paranteze drepte. Valorile `width` (lățime) și `height` (înălțime) specifică dimensiunea ferestrei în pixeli (în afară de `xterm`, unde aceste

valori sunt exprimate ca numere de coloane și rânduri de text). Valorile *xoffset* și *yoffset* sunt de asemenea exprimate în pixeli. Semnificația acestor două numere depinde de aplicație. În *xterm*, de exemplu, o valoare *xoffset* pozitivă indică numărul de pixeli dintre marginea din stânga a ferestrei și marginea din stânga a ecranului. Pe de altă parte, o valoare *xoffset* negativă specifică numărul de pixeli dintre marginea din dreapta a ferestrei și marginea din dreapta a ecranului. Similar, valori pozitive, respectiv negative ale parametrului *yoffset* indică deplasarea față de marginile de sus, respectiv de jos ale ferestrei.

De exemplu, pentru inserția unei ferestre *xterm* lată de 80 de caractere și înaltă de 25 de linii în colțul din dreapta-sus al ecranului, cu un spațiu de 16 pixeli între cadrul ferestrei și marginile din dreapta și din stânga ale ecranului, se tastează comanda:

```
xterm -geometry 80x25-16+16 &
```

Opțiuni pentru controlul aspectului ferestrei

Multe alte opțiuni în linie de comandă determină modul de prezentare al unei ferestre de aplicație. Aceste opțiuni specifică culoarea de prim-plan (-fg), culoarea de fond (-bg), culoarea marginii (-bd) și lățimea marginii (-bw), în pixeli. Culoarele pot fi specificate folosind numele care apar în fișierul `/usr/lib/X11/rgb.txt`. Pentru a lansa programul *xterm* cu galben pe fond albastru ultramarin, folosiți următoarea comandă:

```
xterm -bg navy -fg yellow &
```

Dacă numele unei culori include spații înglobate, atunci introduceți numele între semnele citării. Pentru a specifica *light blue* (albastru deschis) drept culoare de fond pentru *xterm*, folosiți următoarea comandă:

```
xterm -bg "light blue" &
```

Așa cum s-a arătat în secțiunea „Culoarea ecranului” din acest capitol, culoarea mai poate fi specificată drept număr hexazecimal cu șase cifre hexazecimale. Cifrele care compun valoarea RGB hexazecimală a unei culori sunt interpretate de la stânga la dreapta, unde prima pereche de cifre reprezintă roșul, a doua reprezintă verdele și a treia albastrul. Trebuie să inserați caracterul # în fața numărului hexazecimal, pentru a arăta că specificația culorii folosește formatul hexazecimal. Puteți încerca o culoare bizară pentru fundalul *xterm*, prin utilizarea următoarei comenzi:

```
xterm -bg "#ccb0b0" &
```

Specificarea fontului

Resursa font a unei aplicații controlează modul de prezentare a ieșirii text. Similar culorilor, fonturile sunt specificate prin nume. În X, numele fonturilor sunt extrem de descriptive. În continuare este prezentat un exemplu:

```
-adobe-courier-medium-r-normal--12-120-75-75-m-60-iso8859-1
```

Pentru acest font, adobe este producătorul, courier este familia din care face parte, iar grosimea fontului este medie (*medium*) – fontul poate fi și aldin (*bold*), de exemplu. Valoarea *r* arată că fontul este roman. Un *i* în această poziție semnifică italic, iar un *o* indică oblic. Valoarea *normal* este folosită ca parametru pentru lățimea caracterelor și pentru spațierea între caractere; aici se mai pot folosi valorile *condensed* (condensat), *narrow* (îngust) sau *double* (dublu).

Numerele care urmează după cele două liniuțe (--) indică dimensiunea fontului. 12 indică dimensiunea în pixeli a fontului, iar 120 reprezintă dimensiunea exprimată în zecimi de punct tipografic. Următoarele două numere, 75-75, reprezintă rezoluția pe verticală și pe

orizontală pentru care este proiectat fontul. Litera care urmează rezoluției (m) reprezintă spațierea; această valoare poate fi *m* pentru font monospațiat sau *p* pentru font proporțional. Următorul număr, 60, este lățimea medie a tuturor caracterelor din acest font, măsurată în zecimi de pixel (în acest caz, 6 pixeli). Șirul *iso8859-1* identifică setul de caractere al fontului, conform specificațiilor Organizației Internaționale de Standardizare (ISO). În acest caz, setul de caractere este ISO Latin 1, un superset al setului de caractere ASCII (American Standard Code for Information Interchange).



Sugestie

Când specificați un font într-un fișier resursă, nu este necesar să furnizați întregul nume. Puteți folosi asteriscuri (*) pentru câmpuri care pot lua valori arbitrare. Să presupunem că doriți ca fereastra VT100 din *xterm* să folosească un font courier de 12 puncte de grosime medie. Cu o dispersie judicioasă a asteriscurilor, puteți specifica acest font în fișierul de resurse după cum urmează:

```
*VT100Font: *courier-medium-r-normal-*12*
```

Utilizarea programului de gestionare a ferestrelor fwm2

Un *program de gestionare a ferestrelor* este un client special, care se ocupă de interacțiunile între ferestrele diferitelor programe client afișate pe ecran. Aveți nevoie de un asemenea program pentru a controla amplasamentul și dimensiunea ferestrei fiecărui client. În absența unui program de gestionare a ferestrelor, nu puteți modifica dimensiunea unei ferestre și nici locația acesteia.

Pentru a vedea care este rostul unui program de gestionare a ferestrelor, să considerăm cazul a doi clienți – A și B – care afișează pe același ecran. Nici unul dintre clienți nu cunoaște care sunt necesitățile celuilalt. Să presupunem că rulați clientul B după clientul A și că B preia întregul ecran (fereastra root) ca fereastră de ieșire. În acest moment, fereastra clientului A este acoperită de fereastra lui B și nu aveți nici o modalitate de a ajunge la fereastra A.

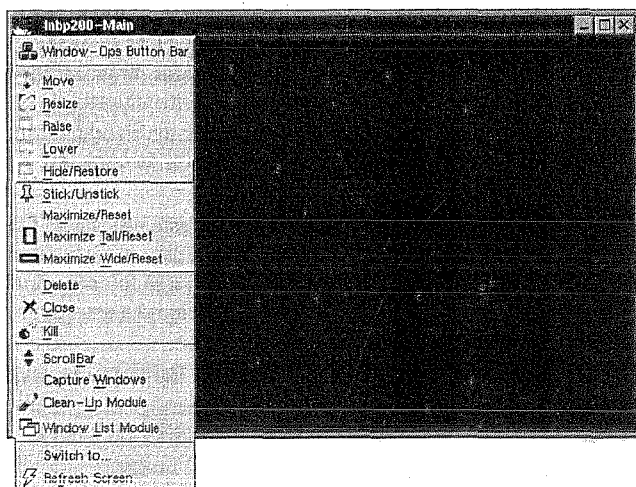
Un program de gestionare a ferestrelor pune la dispoziția utilizatorului modalități de comutare între B și A, chiar și atunci când B se comportă anormal (de exemplu, dacă fereastra lui B ocupă întregul ecran, nu aveți cum să ajungeți la fereastra A fără a folosi un program de gestionare a ferestrelor). Programul de gestionare a ferestrelor *fwm2* adaugă un cadru decorativ la fereastra principală a fiecărei aplicații. Acest cadru vă permite să deplasați și să redimensionați ferestrele. Puteți trage cu mouse-ul un colț al cadrului pentru a restrânge fereastra B. Programul prestabil de gestionare a ferestrelor în Linux (*fwm2*) prezintă un contur al ferestrei, a cărui dimensiune se modifică în conformitate cu deplasarea mouse-ului. Acest contur vă permite să reduceți dimensiunea ferestrei de deasupra, pentru a putea vedea ferestrele plasate sub aceasta. De asemenea, *fwm2* permite utilizarea unui meniu derulant prin efectuarea unui clic pe colțul din stânga-sus al cadrului ferestrei (vezi figura 5.3). Acest meniu conține opțiuni care vă permit să redimensionați o fereastră, să reduceți fereastra la dimensiunea unei pictograme și să părăsiți aplicația.

AnotherLevel și fwm2

Programul *fwm2* se află în directorul `/usr/X11R6/bin`. Versiunea mai veche a programului se numea *fwm*; versiunea 2 conține modificări atât de importante, încât a fost distribuită sub forma unui program nou, denumit *fwm2*.

Figura 5-3

Programul de gestionare a ferestrelor `fwm2` adaugă cadrul și meniul la această fereastră.



Scriptul prestabilit `xclients` din directorul `/etc/X11/xinit` lansează `fwm2` cu setul corespunzător de opțiuni. Aceste opțiuni indică fișierele de configurare pe care `fwm2` trebuie să le încarce. Ceea ce `fwm2` încarcă în mod prestabilit reprezintă un set de fișiere de configurare `fwm2` prestabilite, denumite `AnotherLevel`. Aceste fișiere de configurare se află în directorul `/etc/X11/AnotherLevel` și sunt fișiere text; dacă sunteți curios, le puteți examina folosind comanda `more`. De exemplu, tasteți `more /etc/X11/AnotherLevel/fwm2rc.m4` pentru a consulta principalul fișier de configurare.



Atenție!

Fișierele de configurare `AnotherLevel` sunt foarte complicate – sunt scrise în limbajul macro `m4`. Nu le editați dacă nu cunoașteți acest limbaj. Folosind configurația `AnotherLevel`, la rulare `fwm2` pune la dispoziția utilizatorului un desktop inițial X, așa cum se vede în figura 5.2. Apoi, adaugă un cadru (cu un aspect tridimensional) în jurul fiecărei ferestre de pe ecran.

În următoarele câteva secțiuni vom discuta despre multiple alte facilități ale programului `fwm2`. Când veți citi următoarele paragrafe legate de `fwm2`, nu uitați că aceste descrieri corespund setărilor prestabilite ale programului `fwm2`, așa cum sunt definite de către fișierele de configurare din directorul `/etc/X11/AnotherLevel`. Dacă Red Hat actualizează aceste fișiere într-o versiune ulterioară, aspectul și modul de comportare ale programului `fwm2` se pot modifica. În consecință, nu fiți surprins dacă versiunea `fwm2` din sistemul dumneavoastră nu se comportă exact așa cum este descris în această secțiune.

Suprafața de lucru (desktop) virtuală

Litera `v` din `fwm2` vine de la *virtual*; `fwm2` pune la dispoziția utilizatorului o suprafață de lucru mai mare decât dimensiunile fizice ale ecranului sistemului acestuia. În mod prestabilit, suprafața de lucru virtuală a programului `fwm2` are opt pagini, organizate în două grupe de câte patru (marcate `Desk` și `Apps`). Fiecare pagină are o suprafață egală cu dimensiunea ecranului X. Pentru a asista utilizatorul la navigarea pe suprafața de lucru virtuală, `fwm2` afișează o mică fereastră `pager` într-un colț al ecranului, așa cum se vede în figura 5.4.

Figura 5-4

Puteți naviga pe suprafața de lucru virtuală folosind fereastra `pager`.



Pentru a vă deplasa la o anumită pagină, efectuați clic pe pagina respectivă din fereastra `pager`. Fereastra `pager` este un element constant pe fiecare pagină, deoarece

este o fereastră adevzivă. În `fwm2`, o fereastră adevzivă apare întotdeauna pe fiecare pagină a ecranului virtual. Puteți transforma orice fereastră într-o fereastră adevzivă prin selectarea opțiunii `Stick/Unstick` din meniul aferent ferestrei (vezi figura 5.3).



Secret

Chiar dacă aveți la dispoziție mai multe pagini virtuale de ecran, necesitatea de memorie a programului `fwm2` depinde numai de numărul de ferestre. Dimensiunea suprafeței de lucru virtuale nu are nici o influență asupra cantității de memorie folosite de `fwm2`. Ca atare, puteți folosi facilitatea de suprafață de lucru virtuală a programului `fwm2` fără riscul de a irosi memoria.

Bara de taskuri

O altă caracteristică esențială a programului `fwm2` o reprezintă bara de taskuri stil Windows 95 (vezi figura 5.5), care apare de-a lungul laturii inferioare a ecranului.

Figura 5-5



Programul de gestionare a ferestrelor `fwm2` afișează o bară de taskuri stil Windows 95.

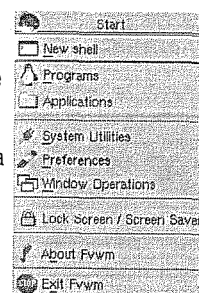
Bara de taskuri afișează un meniu `Start` și butoane pentru fiecare fereastră (pe orice pagină a suprafeței de lucru virtuale). Marginea din dreapta a barei de sarcini afișează data și ora curente.

Meniul Start

Dacă efectuați clic cu butonul stâng de mouse pe butonul `Start` de pe bara de stare, `fwm2` afișează meniul `Start`. Același meniu apare dacă efectuați clic cu butonul stâng de mouse, în timp ce indicatorul de mouse se află oriunde în fereastra rădăcină. De obicei, meniul prezintă articole care lansează o aplicație. Unele dintre articolele de meniu conțin o săgeată; la amplasarea indicatorului de mouse pe un asemenea articol, va apărea un submeniu. Figura 5.6 prezintă meniul `Start` de pe sistemul meu.

Figura 5-6

Efectuați clic cu butonul stâng de mouse pe fereastra `root` pentru a vizualiza meniul `Start`.



Trebuie să explorați toate articolele din meniul `Start`, pentru a vedea toate operațiile pe care le puteți efectua pornind de la acest meniu. În particular, trebuie să deplasați mouse-ul deasupra articolului `Preferences` (preferințe) pentru a vedea submeniul (vezi figura 5.7) care prezintă opțiunile utile pentru modificarea aspectului suprafeței de lucru.

Așa cum se vede în figura 5.7, submeniul `Preferences` vă permite să modificați fundalul ferestrei `root`, forma cursorului, precum și multe alte elemente ale suprafeței de lucru.

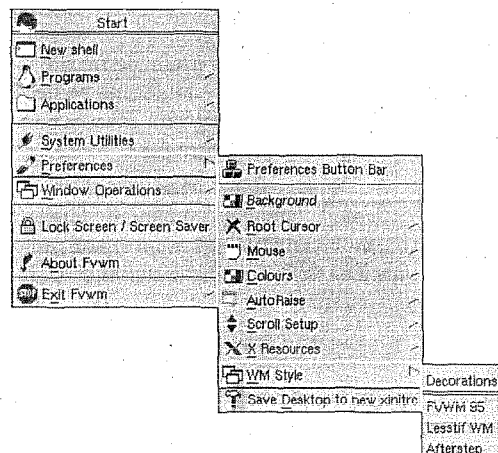
Alte stiluri de prezentare a suprafeței de lucru

Unul dintre cele mai ciudate articole din submeniul `Preferences` este opțiunea `WM Style`, selectată în figura 5.7. Puteți modifica întregul stil al suprafeței de lucru prin selectarea

articolelor din submeniul respectiv. Stilul prestabilit este FVWM 95, care reproduce (într-o oarecare măsură) suprafața de lucru Windows 95.

Figura 5-7

Submeniul Preferences vă permite să modificați modul de prezentare al suprafeței de lucru.



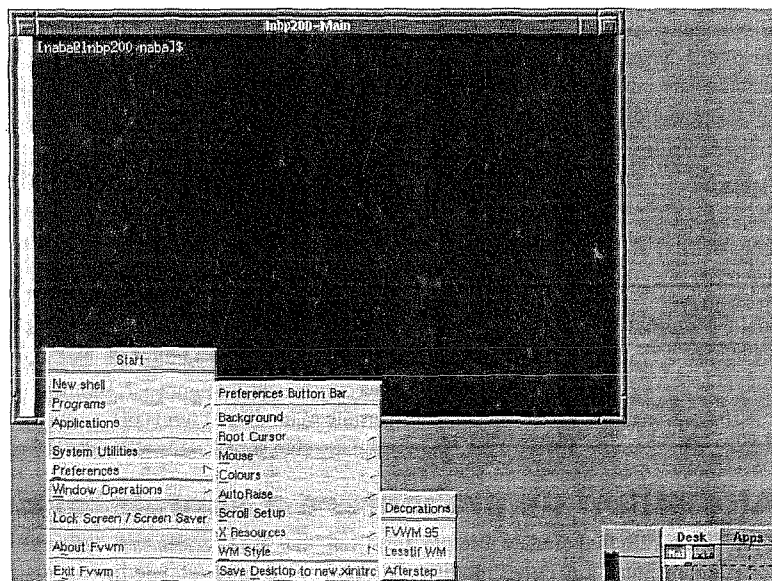
Submeniul WM Style mai include două opțiuni:

- LessTif WM pune la dispoziția utilizatorului un aspect stil Motif.
- Afterstep este un aspect stil NextStep.

Pentru a încerca aspectul Motif, selectați opțiunea LessTif din submeniul WM prezentat în figura 5.7. Bara de sarcini stil Windows 95 dispare și elementele decorative ale ferestrei se transformă în cele ale stilului Motif, așa cum se vede în figura 5.8.

Figura 5-8

O suprafață de lucru în stil Motif, cu programul de gestionare a ferestrelor fvwm2.

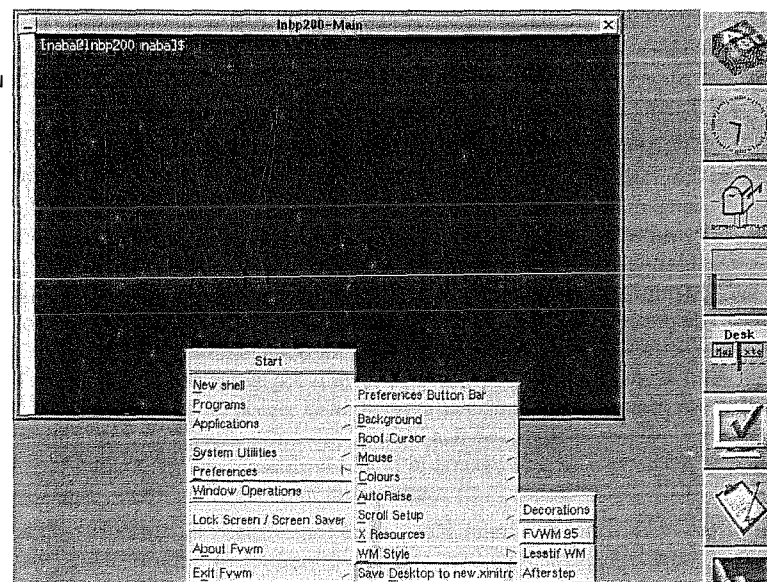


Mai puteți ajunge la meniul Start (vezi figura 5.8): pur și simplu efectuați clic pe butonul stâng de mouse, cu indicatorul plasat oriunde în fereastra root. Dacă doriți să reveniți la stilul de suprafață de lucru Windows 95 sau să încercați stilul Afterstep, deplasați indicatorul spre articolul Preferences și apoi spre WM Style. Din meniul WM Style, selectați stilul de suprafață de lucru preferat.

De exemplu, dacă optați pentru stilul Afterstep, suprafața de lucru capătă aspectul prezentat în figura 5.9.

Figura 5-9

O suprafață de lucru Afterstep, cu programul de gestionare a ferestrelor fvwm2.



Elementele decorative ale ferestrei apar în mod diferit, iar marginea din dreapta are o coloană cu butoane, care permit efectuarea unei operații specifice, precum lansarea unui editor de texte sau deschiderea unei ferestre terminal.



Sugestie

După cum se vede în figura 5.9, puteți ajunge la meniul Start dacă efectuați clic pe butonul stâng de mouse, cu indicatorul de mouse amplasat oriunde în fereastra root. Selectând Preferences⇒WM Style⇒FVWM 95, puteți reveni la suprafața de lucru stil Windows 95.

Rezumat

Începător sau experimentat, puteți beneficia de o interfață cu utilizatorul care folosește indicarea cu mouse-ul și efectuarea de clic. Acest capitol vă prezintă modul de configurare Linux pentru a-l lansa într-un mod grafic, pentru ca utilizatorul să se poată menține într-un mediu grafic.

Din lectura acestui capitol ați învățat următoarele:

- Sistemul X Window System include un program denumit X Display Manager (xdm) care pune la dispoziția utilizatorului un prompt grafic de intrare în sistem. Când veți înțelege modul de lansare a proceselor în Linux, veți lansa xdm la fiecare încărcare a sistemului de operare.

- După încărcarea sistemului de operare Linux, un proces denumit `init` lansează toate celelalte procese. Setul exact de procese inițiale depinde de nivelul de rulare, care de regulă este un număr cuprins între 0 și 6. Fișierul `/etc/inittab` specifică procesele care sunt lansate în fiecare nivel de rulare.
- Fișierul `/etc/inittab` specifică faptul că programul X Display Manager (`xm`) rulează la nivelul de rulare 5. Ca atare, puteți lansa `xm` cu comanda `init 5`. Această comandă este o bună metodă de testare a programului `xm`.
- Orice aplicație X poate fi personalizată cu ușurință. Aspectele adaptabile ale unei aplicații X sunt cunoscute sub numele de *resurse*. Puteți specifica valorile resurselor în fișierul `.xresources` din directorul dumneavoastră de bază. Prin intermediul resurselor, puteți specifica parametri precum culoarea de fond, culoarea de prim-plan și fonturile.
- Programul de gestionare a ferestrelor controlează o bună parte din aspectul și modul de comportare a mediului grafic X Window System. Programul de gestionare a ferestrelor `fvwm2` este foarte adaptabil; în Linux Red Hat, configurația sa este controlată de fișierele de configurare `AnotherLevel`, plasate în directorul `/etc/X11/AnotherLevel`. Meniurile programului `fvwm2` vă permit personalizarea mai multor aspecte ale modului de prezentare a suprafeței de lucru.

Capitolul 6

Secretele comenzilor Linux

În acest capitol:

- Veți înțelege shellurile: interpretoarele de comenzi din Linux
- Veți examina interpretorul Bash (Bourne Again Shell)
- Veți înțelege structura de directoare a sistemului de fișiere Linux
- Veți folosi comenzi Linux care prelucrează fișiere
- Veți scrie scripturi de interpretor
- Veți realiza automatizarea unor activități de rutină cu un script de interpretor
- Veți înțelege noțiuni de bază ale programării în limbajul Perl

Ați văzut în capitolul precedent cum puteți lucra într-un mediu grafic sub Linux. Din nefericire, nu chiar toate problemele pot fi rezolvate dintr-un mediu grafic. Nu este imposibil să proiectați o interfață grafică care să vă permită să executați majoritatea activităților de rutină, dar sistemul Linux nu include o interfață grafică atât de cuprinzătoare. În consecință, chiar și în mediul grafic din X, trebuie deseori să lucrați în ferestre `xterm`, în care puteți introduce comenzi Linux pentru a efectua anumite activități.

În Linux dispuneți de o varietate de interpretoare de comenzi, numite shelluri. Acest capitol vă prezintă interpretorul *Bash* (Bourne Again Shell), care este shellul prestabilit în Linux. Veți învăța câteva dintre cele mai importante comenzi shell și veți vedea cum puteți scrie *scripturi de interpretor* simple. Scripturile de interpretor reprezintă colecții de comenzi shell adunate în fișiere. În ceea ce privește elaborarea de scripturi, printre administratorii de sisteme UNIX este foarte răspândit un limbaj numit Perl. Cum dumneavoastră sunteți probabil administratorul propriului sistem UNIX, acest capitol vă oferă și o scurtă introducere în limbajul Perl.



Referință
încrucișată

Am descris limbajul Tcl/Tk, un alt limbaj foarte răspândit pe care îl puteți utiliza pentru elaborarea de aplicații într-o interfață grafică, în capitolul 8.

Interpretorul Bash

Dacă ați utilizat sistemul MS-DOS, cunoașteți desigur `COMMAND.COM`, interpretorul de comenzi din DOS, programul care afișează oribilul prompt `C:\>`. Sistemul Linux oferă un interpretor de comenzi similar cu `COMMAND.COM`-ul din DOS. În UNIX, interpretorul de comenzi este denumit tradițional shell. Shellul inițial UNIX era shell-ul Bourne, iar programul executabil asociat era numit `sh`. Shellul prestabilit utilizat în Linux este Bash, iar numele programului executabil asociat este `bash` (acesta se găsește în directorul `/bin`). Bash este compatibil cu shellul inițial, `sh`, dar conține în plus multe facilități ale altor interpretoare binecunoscute, cum ar fi C sau Korn. De exemplu, Bash vă permite să reutilizați comenzi pe care le-ați introdus anterior; shellul completează de asemenea numele comenzilor introduse doar parțial.

Scopul urmărit de shellul Bash este de a afișa un prompt și de a executa comenzile pe care le introduceți de la tastatură. Unele dintre aceste comenzi, cum ar fi `cd` (change directory – modificare director) sau `pwd` (print working directory – afișare director de lucru) sunt încorporate direct în Bash. Multe alte comenzi, cum ar fi `cp` (copy) sau `ls` (list directory – listă director) reprezintă programe separate (ceea ce înseamnă că fișierele care implementează aceste comenzi se găsesc în anumite directoare din sistemul de fișiere). Ca utilizator, nu este necesar să știți dacă o anumită comandă este încorporată în interpretor sau reprezintă un program executabil separat.

În plus față de comenzile standard Linux, Bash poate executa orice program memorat într-un fișier executabil. Bash poate executa de asemenea orice script interpretor (un fișier text care conține una sau mai multe comenzi). După cum veți vedea ulterior în secțiunea „Scripturi interpretor în Bash” din acest capitol, puteți utiliza și comenzi încorporate ale interpretorului pentru a scrie programe (denumite scripturi shell).

Următoarele câteva secțiuni vă ajută să vă formați o imagine despre diferitele facilități oferite de un interpretor, de la sintaxa generală a comenzilor și până la bazele programării în interpretor. După parcurgerea acestei introduceri, puteți afla mai multe detalii despre aceste subiecte în secțiunile ulterioare.



Observație

În acest capitol, vom presupune că utilizați interpretorul Bash, deoarece acesta este shellul selectat în mod prestabilit după instalarea sistemului Linux de pe CD-ROM-ul atașat lucrării.

Sintaxa comenzilor

Din cauză că interpretorul interpretează ceea ce tastezi, cunoașterea modului în care acesta prelucrează textul pe care îl introduceți este foarte importantă. Toate comenzile interpretor respectă următorul format general:

comanda optiune1 optiune2 ... optiuneN

Linia pe care puteți introduce comenzi este uzual denumită *linie de comandă*. Pe o linie de comandă, puteți introduce o comandă urmată de una sau mai multe opțiuni (sau parametri), denumite uzual *opțiuni în linia de comandă* (sau *parametri în linia de comandă*).

O regulă de bază este aceea că trebuie să separați numele comenzii de opțiuni printr-un spațiu sau caracter de tabulare. Opțiunile trebuie de asemenea separate între ele prin spațiu sau caracter de tabulare. Dacă doriți să utilizați o opțiune care conține spații, acea opțiune trebuie inclusă între ghilimele. Pentru a căuta un nume într-un fișier de parole, de exemplu, trebuie introdusă următoarea comandă `grep`:

```
grep "Naba Barkakati" /etc/passwd
```

Atunci când comanda `grep` afișează linia care conține numele, aceasta are forma:

```
naba:Lgb7s0ywtVswx:500:100:Naba Barkakati:/home/naba:/bin/bash
```

Această linie conține multe informații utile, cel mai interesant (din punctul de vedere al discuției de față) fiind câmpul care urmează după ultimul caracter „două puncte” (:). Acest câmp conține numele interpretorului rulat de către utilizator.

Numărul și formatul opțiunilor din linia de comandă depind de fiecare comandă în parte. Când veți învăța mai multe despre comenzi, veți vedea că opțiunile care determină comportamentul unei anumite comenzi sunt de forma `X`, unde `X` este un singur caracter.

Dacă o comandă este prea lungă pentru a încăpea pe o singură linie, puteți apăsa tasta backslash (\), urmată de Enter. Puteți apoi continua să introduceți comanda pe linia următoare. Puteți de asemenea concatena mai multe comenzi pe o singură linie; utilizați caracterul punct și virgulă (;) ca separator între comenzi. De exemplu, la re-generarea nucleului Linux (prezentată în capitolul 2) puteți efectua trei activități succesive tastând următoarele trei comenzi pe o singură linie:

```
make dep; make clean; make boot
```

Combinatii de comenzi

Sistemul Linux implementează filosofia UNIX de a pune la dispoziția utilizatorului o mulțime de comenzi simple. Puteți însă să combinați aceste comenzi pentru a crea comenzi mai sofisticate. Să presupunem că doriți să determinați dacă în directorul `/dev` există un dispozitiv numit `sbpcd`. Acest fișier vă este necesar dacă doriți să instalați o unitate CD-ROM Sound Blaster Pro. Puteți utiliza comanda `ls /dev` pentru a lista conținutul directorului `/dev`, după care puteți vedea dacă `sbpcd` apare în listă. Din păcate, directorul `/dev` conține multe fișiere și este dificil să localizați liniile care conțin `sbpcd` în cuprinsul numelui. Puteți însă combina comanda `ls` cu `grep`, obținând o nouă comandă care efectuează exact ceea ce dorește utilizatorul:

```
ls /dev | grep sbpcd
```

Shellul va trimite ieșirea comenzii `ls` (adică listingul directorului `/dev`) ca intrare pentru comanda `grep`, care caută șirul de caractere `sbpcd`. Bara verticală (|) este cunoscută sub numele de caracter de redirectare (pipe), deoarece se comportă ca o legătură între două programe: ieșirea primului program devine intrare pentru cel de-al doilea.



Observație

Majoritatea comenzilor Linux sunt proiectate într-un mod care permite ca ieșirea lor să fie furnizată ca intrare pentru altă comandă. Pentru aceasta, este suficient să concatenați cele două comenzi, separându-le prin caracterul de redirectare.

Redirectarea intrărilor și ieșirilor

Comenzile Linux care sunt proiectate pentru a lucra împreună au un atribut comun: toate citesc date de la *intrarea standard* (care este de obicei tastatura) și scriu rezultate la *ieșirea standard* (de regulă, ecranul). Dacă doriți ca o comandă să citească date dintr-un fișier, îi puteți redirecta intrarea standard, astfel încât să citească date din acel fișier. Similar, pentru a salva rezultatele unei comenzi într-un fișier, puteți redirecta ieșirea standard într-un fișier. Aceste facilități oferite de interpretor sunt denumite *redirectarea intrării și ieșirii*.

Utilizând comanda următoare, puteți căuta printre fișierele din directorul `/usr/include` pe cele care conțin șirul de caractere `typedef`, salvând apoi lista acestor fișiere într-un fișier numit `typedef.out`:

```
grep typedef /usr/include/* > typedef.out
```

Această comandă ilustrează și o altă facilități a interpretorului Bash. Atunci când utilizați un asterisc (*), Bash îl înlocuiește cu lista tuturor numelor fișierelor din directorul specificat. Prin urmare, `/usr/include/*` semnifică mulțimea tuturor fișierelor din directorul `/usr/include`.

Programe de shell

Dacă nu sunteți un programator, probabil că resimțiți o oarecare teamă în ceea ce privește programarea. Programarea în interpretor este însă destul de simplă, constând doar în

memorarea unei secvențe de comenzi într-un fișier. Este posibil chiar să construiți un program de shell util care să aibă o singură comandă.

La elaborarea lucrării de față, a fost necesară capturarea ecranelor din X Window System pentru a le utiliza ca figuri. S-a utilizat programul de capturare a ecranelor în X, `xwd`, pentru a memora imaginile reprezentând ecrane în format X Window Dump (XWD). Echipa de tehnoredactare a cărții avea însă nevoie de acele ecrane în format PCX. Ca atare, s-a utilizat setul de instrumente Portable Bitmap (PBM) pentru conversia imaginilor din format XWD în format PCX. Pentru conversia unui fișier este necesară rularea a două programe și ștergerea unui fișier temporar, după cum urmează:

```
xwdtopnm < file.xwd > file.ppm
ppmtopcx < file.ppm > file.pcx
rm file.ppm
```

Aceste comenzi presupun că programele `xwdtopnm` și `ppmtopcx` există într-unul din directoarele care fac parte din variabila de mediu `PATH`. Cele două programe de mai sus, `xwdtopnm` și `ppmtopcx` fac parte din setul de instrumente PBM.

După conversia câtorva fișiere XWD în format PCX, vă veți plictisi să tastați aceeași secvență de comenzi pentru fiecare fișier în parte. În acest moment, puteți deschide un fișier numit `topcx`, în care să scrieți liniile:

```
xwdtopnm < $1.xwd > $1.ppm
ppmtopcx < $1.ppm > $1.pcx
rm $1.ppm
```

Acest fișier trebuie acum să devină executabil. Pentru aceasta, vom folosi comanda:

```
chmod +x topcx
```

Comanda `chmod` vă permite să modificați permisiunile asociate unui fișier. Una dintre aceste permisiuni stabilește dacă fișierul este executabil sau nu. Opțiunea `+x` înseamnă că doriți să marcați fișierul ca fiind executabil. Această operație este necesară, întrucât Bash va rula numai fișiere executabile.

În fine, puteți converti fișierul `figure1.xwd` în `figure1.pcx` utilizând comanda:

```
topcx figure1
```

Fișierul `topcx` este numit *program de interpretor*. La rularea acestui program cu comanda `topcx figure1`, shellul substituie fiecare apariție a șirului de caractere `$1` din program cu parametrul `figure1`.

În concluzie, programele interpretor sunt necesare pentru a determina sistemul Linux să execute automat activitățile repetitive.

Iată în continuare un alt exemplu interesant de program de interpretor. Să presupunem că uneori utilizați fișiere text în MS-DOS în sistemul dumneavoastră Linux. Deși credeți că puteți utiliza fișiere text în orice sistem de operare fără nici un fel de probleme, există un impediment: sistemul DOS marchează sfârșitul unei linii printr-o succesiune de două caractere, retur de car (Carriage Return – CR), urmat de salt la linie nouă (Line Feed – LF), în vreme ce în Linux (și în alte sisteme UNIX), sfârșitul liniilor se marchează doar prin caracterul salt la linie nouă. Ca rezultat, dacă utilizați editorul `vi` pentru a deschide un fișier text din DOS, veți observa caracterul special `^M` la sfârșitul fiecărei linii. Secvența `^M` se citește Ctrl-M, fiind utilizată în locul caracterului retur de car.

În sistemul dumneavoastră Linux, puteți elimina cu ușurință caracterele retur de car dintr-un fișier text DOS utilizând comanda `tr` cu opțiunea `-d`. În esență, pentru conversia unui fișier text DOS `nume_fisier.dos` în fișierul text Linux `nume_fisier.linux`, puteți utiliza comanda:

```
tr -d '\015' < nume_fisier.dos > nume_fisier.linux
```

În această comandă, `'\015'` reprezintă codul ASCII pentru caracterul retur de car în notație octală.



Observație

Comanda `tr` poate fi utilizată pentru a înlocui sau șterge anumite caractere din fișierul de intrare. Atunci când utilizați comanda `tr` cu opțiunea `-d`, aceasta șterge toate aparițiile unui anumit caracter din datele de intrare. După opțiunea `-d`, trebuie să specificați caracterul care va fi șters. Ca și multe alte utilitare UNIX, `tr` citește intrarea standard și scrie la ieșirea standard. După cum ați observat în exemplul de mai sus, trebuie să utilizați redirectarea intrării și ieșirii pentru a determina comanda `tr` să șteargă toate aparițiile unui caracter într-un fișier și apoi să salveze ieșirea în alt fișier.

Dacă nu doriți să vă amintiți toate aceste informații de fiecare dată când aveți de convertit un fișier text DOS în format UNIX, salvați linia următoare într-un fișier numit `dos2unix`:

```
tr -d '\015' < $1 > $2
```

Acordați apoi acestui fișier dreptul de execuție, utilizând comanda:

```
chmod +x dos2unix
```

Gata! Aveți acum un program de interpretor numit `dos2unix`, care convertește un fișier text DOS în format UNIX. Dacă aveți partiția DOS montată în directorul `/dos`, puteți încerca programul interpretor `dos2unix` cu următoarea comandă:

```
dos2unix /dos/autoexec.bat aexec.bat
```

Această comandă creează un fișier numit `aexec.bat` în directorul curent. Dacă deschideți acest fișier cu editorul `vi`, nu veți vedea nici un caracter special `^M` la sfârșitul liniilor.



Observație

Dacă aveți experiență în MS-DOS, veți observa că scripturile de shell sunt foarte asemănătoare cu fișierele de comenzi (batch) din MS-DOS. Cu excepția unor diferențe de sintaxă, scripturile de shell sunt similare cu fișierele de comenzi DOS.

Variabile de mediu

Comenzile de interpretor și alte comenzi Linux necesită anumite informații pentru a funcționa corespunzător. Dacă tastați o comandă care nu este încorporată în interpretor, acesta trebuie să localizeze un fișier executabil (al cărui nume trebuie să coincidă cu numele comenzii introduse). Interpretorul trebuie să știe în ce directoare caută astfel de fișiere executabile. În mod similar, un editor de texte cum este `vi` trebuie să cunoască tipul terminalului (chiar dacă acesta este `xterm`, care emulează un terminal într-o fereastră X).

O modalitate prin care puteți comunica aceste informații unui program este prin intermediul opțiunilor din linia de comandă. Dacă optați pentru această metodă, va trebui însă ca de fiecare dată când lansați un program să introduceți foarte multe opțiuni. Sistemul UNIX oferă o soluție elegantă la această problemă: variabilele de mediu.

O *variabilă de mediu* este un nume căruia i se asociază un șir de caractere. Într-un anumit sistem, variabila de mediu `PATH` poate fi definită după cum urmează:

```
PATH=/usr/local/bin:/bin:/usr/sbin:/usr/X11R6/bin
```

Șirul de caractere din partea dreaptă a semnului egal reprezintă valoarea variabilei de mediu `PATH`. Prin convenție, variabila de mediu `PATH` reprezintă o secvență de nume de directoare, fiecare dintre acestea fiind separat de cel precedent prin caracterul două puncte (`:`). Punctul prezent în această listă reprezintă de asemenea un director – directorul curent.

Atunci când interpretorul are de căutat un fișier, caută prin directoarele cuprinse în variabila de mediu PATH. Căutarea în aceste directoare se face în ordinea apariției lor în cadrul variabilei. În consecință, dacă două programe au același nume, shell-ul îl va executa pe cel pe care îl găsește primul.

Într-o manieră similară celei în care interpretorul utilizează variabila de mediu PATH, un editor cum este vi poate utiliza variabila de mediu TERM pentru a determina modul de afișare a fișierului pe care îl editează. Pentru a vedea care este valoarea curentă a variabilei TERM, tasteți comanda următoare la promptul shellului:

```
echo $TERM
```

Dacă ați tastat această comandă într-o fereastră xterm, veți obține răspunsul:

```
xterm
```

Pentru a defini o variabilă de mediu în Bash, trebuie să utilizați sintaxa următoare:

```
export NAME=Value
```

Aici, *NAME* este numele variabilei de mediu, iar *Value* este șirul de caractere care reprezintă valoarea variabilei. Prin urmare, pentru a atribui variabilei TERM valoarea xterm, puteți utiliza comanda:

```
export TERM=xterm
```

Pentru o variabilă de mediu cum este PATH, este mai degrabă necesar să adăugați un nou director la definiția existentă, decât să redefiniți complet variabila. Exemplul următor vă arată modul cum puteți realiza această operație:

```
export PATH="$PATH:/usr/games"
```

Această comandă adaugă șirul de caractere `/usr/games` la definiția curentă a variabilei de mediu PATH. Efectul este adăugarea directorului `/usr/games` în lista directorilor din PATH.

PATH și TERM sunt numai două dintre variabilele de mediu mai frecvent utilizate. Tabelul 6-1 cuprinde câteva dintre variabilele de mediu utile din Bash.

Tabelul 6-1 Variabile de mediu utile în Bash

Variabilă de mediu	Semnificație
DISPLAY	Numele ecranului pe care X Window System afișează în mod curent (de regulă, acesta are valoarea <code>:0.0</code>)
HOME	Directorul dumneavoastră personal
LOGNAME	Numele dumneavoastră de login
PATH	Lista directorilor în care interpretorul caută programe
PS1	Promptul shellului (cel prestabilit este <code>bash\$</code> , pentru utilizatorii diferiți de <code>root</code> ; pentru <code>root</code> , acesta este <code>bash#</code>).
SHELL	Interpretorul pe care îl utilizați (<code>SHELL=/bin/bash</code> dacă utilizați chiar Bash)
TERM	Tipul terminalului utilizat

Procese

De fiecare dată când interpretorul interpretează o comandă pe care ați tastat-o, lansează de fapt un nou *proces*. Interpretorul însuși este un proces, ca de altfel toate scripturile și programele pe care acesta le execută. Un exemplu de astfel de program este programul de gestionare a ferestrelor `fvwm2`. Puteți vizualiza o listă a proceselor utilizând comanda `ps`. De exemplu,

atunci când tasteți `ps`, Bash afișează setul de procese care rulează în momentul curent. În continuare, iată un exemplu de raport care se poate obține dacă tasteți comanda `ps` într-o fereastră `xterm`:

```
ps
PID TTY STAT TIME COMMAND
344 1 S 0:00 /bin/login - - naba
352 1 S 0:00 -bash
363 1 S 0:00 sh /usr/X11R6/bin/startx
364 1 S 0:00 xinit /usr/X11R6/lib/X11/xinit/xinitrc - -
367 1 S 0:00 fvwm2 -cmd FvwmM4 -debug /etc/X11/AnotherLevel/fvwm2rc.m4
436 1 S 0:00 /usr/X11R6/lib/X11/fvwm2//FvwmTaskBar 9 4 /tmp/fvwmrca00376
437 1 S 0:00 /usr/X11R6/lib/X11/fvwm2//FvwmButtons 11 4 /tmp/fvwmrca00376
440 1 S NO:00 xload -nolabel -geometry 32x20+0+0 -bg grey60 -update 5
441 1 S 0:00 /usr/X11R6/lib/X11/fvwm2//FvwmPager 11 4 /tmp/fvwmrca00376
442 p0 S 0:00 bash
501 p0 R 0:00 ps
```

În formatul prestabilit de afișare, coloana *COMMAND* arată comenzile care au creat procesele. Această listă include printre procese interpretorul `bash` și chiar comanda `ps`. Alte procese aflate în rulare sunt programele lansate de interpretor atunci când s-a tastat `startx` pentru a se încărca sistemul X. În particular, această listă include și procesul `fvwm2` (programul de gestionare a ferestrelor).

Comanda `ps` fără opțiuni nu arată însă toate procesele aflate în rulare într-un sistem Linux. Comanda listează numai acele comenzi pe care le-ați lansat, fie direct, fie indirect (prin intermediul unor scripturi care s-au executat când ați intrat în sistem). Pentru a vizualiza întregul set de procese, utilizați opțiunea `a` a comenzii `ps` împreună cu opțiunea `x`, după cum urmează:

```
ps ax
```

Nu prezentăm aici rezultatul acestei comenzi, deoarece se află în rulare un număr mare de procese, chiar și atunci când sunteți singurul utilizator al sistemului. Estimativ, numărul proceselor din lista afișată este cuprins între 30 și 45.



Secret

Examinând rezultatul afișat de comanda `ps`, veți observa că prima coloană are antetul *PID* și cuprinde un număr pentru fiecare proces. Abrevierea *PID* vine de la *process ID* (identificator de proces), care este un număr secvențial atribuit de nucleul sistemului Linux. Dacă examinați rezultatul comenzii `ps ax`, veți observa că procesul `init` este primul care se execută, având asociat un număr *PID* egal cu 1. Din acest motiv, `init` este denumit *matricea tuturor proceselor*.



Sugestie

Identificatorul de proces este util atunci când doriți să opriți forțat un proces eronat. Examinați rezultatul comenzii `ps ax` și notați *PID*-ul procesului respectiv. Utilizați apoi comanda `kill` dându-i ca parametru acest număr. Pentru a opri, de exemplu, procesul cu numărul 123, tasteți `kill -9 123`.



Secret

Sistemele UNIX, deci și Linux, utilizează semnale pentru a înștiința un proces de apariția unui anumit eveniment. Comanda `kill` vă permite să trimiteți un semnal unui anumit proces (identificat prin numărul de proces). Opțiunea `-9` a comenzii `kill` indică semnalul care va fi trimis; 9 este numărul semnalului `SIGKILL`, care determină terminarea necondiționată a procesului.

Comenzi în fundal și terminale virtuale

În sistemul MS-DOS, trebuie să așteptați terminarea fiecărei comenzi înainte de a o putea introduce pe următoarea. (Puteți continua să tasteți, dar sistemul MS-DOS poate reține în bufferul său intern numai câteva caractere.) Sistemul Linux poate însă gestiona simultan

activități multiple. Singura problemă este aceea că terminalul sau consola vor fi blocate până la terminarea comenzii.

Dacă lucrați într-o fereastră `xterm` și comanda pe care ați introdus-o durează foarte mult, puteți deschide o altă fereastră `xterm` în care să introduceți comenzi pentru a vă continua lucrul fără întreruperi. Dacă însă lucrați în mod text, iar comanda pe care ați introdus-o durează foarte mult, aveți nevoie de o altă modalitate de acces la sistem.

Puteți continua să lucrați în mai multe moduri în timp ce sistemul dumneavoastră Linux execută o activitate mare consumatoare de timp:

- Puteți lansa comenzile cronofage *în fundal*, ceea ce înseamnă că interpretorul va lansa procesul care execută comanda, revenind însă imediat pentru a accepta alte comenzi. Interpretorul nu așteaptă în acest caz terminarea comenzii; comanda rulează ca proces distinct în fundal. Pentru a lansa un proces în fundal, este suficient să tastați un ampersand (&) la sfârșitul liniei de comandă. Spre exemplu, dacă dorim să rulăm scriptul shell `topcx` pentru a converti un fișier mare numit `image1.xwd` în format PCX, putem rula scriptul în fundal utilizând comanda:

```
topcx image1 &
```

- Dacă o anumită comandă (pe care nu ați lansat-o în fundal) durează mai mult decât v-ați fi așteptat, tastați `Ctrl-Z` pentru a o opri; tastați apoi `bg` pentru a rula acel proces în fundal.

- Utilizați facilitatea de *terminal virtual* oferită de Linux. Chiar dacă sistemul Linux are numai un singur terminal fizic (prin terminal înțelegem combinația dintre monitor și tastatură), utilizatorul are impresia că dispune de terminale multiple. Ecranul text inițial este pe primul terminal virtual. Tastați `Alt-F2` pentru a vă deplasa la cel de-al doilea terminal virtual, `Alt-F3` pentru cel de-al treilea ș.a.m.d. Din X Window System, trebuie să tastați `Ctrl-Alt-F1` pentru a ajunge pe primul terminal virtual, `Ctrl-Alt-F2` pentru cel de-al doilea etc.



Sugestie

Pentru a ajunge înapoi pe consola X, tastați `Ctrl-Alt-F7`. Puteți utiliza unul dintre terminalele virtuale pentru a distruge procesele care pot determina blocarea ecranului X (care se produce atunci când mouse-ul nu mai răspunde).

Completarea comenzilor în Bash

Multe comenzi iau ca parametru numele unui fișier. Dacă doriți să vizualizați conținutul fișierului `/etc/X11/XF86Config`, trebuie să tastați comanda:

```
more /etc/X11/XF86Config
```

Această linie determină comanda `more` să afișeze conținutul fișierului `/etc/X11/XF86Config` ecran cu ecran. Pentru comenzile care au ca parametru numele unui fișier, Bash pune la dispoziție o facilități care vă permite să tastați un nume ceva mai scurt. Este suficient să tastați un minimum – doar primele caractere – pentru a identifica fișierul dorit în mod unic în directorul în care se găsește.

Pentru a încerca un exemplu, tastați `more /etc/X11/XF`, dar nu apăsați încă `Enter`, ci `Tab`. Interpretorul Bash va completa automat numele fișierului, astfel încât comanda va deveni `more /etc/X11/XF86Config`. Apăsați acum tasta `Enter` pentru a rula comanda.



Sugestie

Ori de câte ori trebuie să introduceți numele unui fișier, tastați `Tab` după ce ați introdus primele câteva caractere ale numelui. Interpretorul Bash va putea, în majoritatea cazurilor, să completeze numele, deci nu este cazul să tastați numele întreg. Dacă nu ați introdus încă suficiente caractere pentru a identifica fișierul în mod unic, interpretorul Bash va emite un sunet. Tastați alte câteva caractere și apăsați din nou `Tab`.

Caractere de înlocuire

Un alt mod de a evita introducerea prea multor caractere este utilizarea caracterelor de înlocuire, cum ar fi asteriscul (*) sau semnul de întrebare (?), care pot substitui zero sau mai multe caractere din orice șir de caractere. În MS-DOS, se pot copia toate fișierele din directorul curent în unitatea A utilizând comanda `COPY *.* A:`. Interpretorul Bash acceptă astfel de caractere de înlocuire în numele fișierelor. În realitate, în Bash puteți utiliza mai multe caractere de înlocuire decât în MS-DOS.

Shellul Bash suportă trei tipuri de caractere de înlocuire:

- Caracterul asterisc (*) substituie zero sau mai multe caractere din numele oricărui fișier. Prin urmare, * reprezintă toate fișierele dintr-un director.

- Semnul de întrebare (?) se potrivește cu orice caracter.

- Un set de caractere cuprinse între paranteze substituie orice caracter din set. Spre exemplu, șirul de caractere `[xx]*` poate înlocui orice nume de fișier care începe cu una din literele `x` sau `x`.

Caracterele de înlocuire sunt utile atunci când doriți să executați o anumită acțiune asupra unui grup de fișiere. Pentru a copia toate fișierele din directorul numit `/mnt` în directorul curent, puteți introduce comanda:

```
cp /mnt/* .
```

Shellul Bash va înlocui caracterul de înlocuire * cu numele tuturor fișierelor din directorul `/mnt`. Punctul care încheie comanda reprezintă directorul curent.

Puteți utiliza asteriscul în alte părți ale numelui unui fișier pentru a selecta un grup de fișiere mai specific. Să presupunem că doriți să utilizați comanda `grep` pentru a căuta șirul de caractere `typedef struct` în toate fișierele din directorul `/usr/include` care îndeplinesc în plus următoarele condiții:

- Numele fișierului începe cu `s`.

- Numele fișierului se termină cu `.h`.

Specificația `s*.h` substituie toate numele fișierelor care satisfac aceste criterii. Prin urmare, puteți executa căutarea utilizând comanda următoare:

```
grep "typedef struct" /usr/include/s*.h
```

Șirul de caractere care trebuie căutat de comanda `grep` conține un spațiu, prin urmare trebuie inclus între ghilimele. Astfel, ne asigurăm că shellul Bash nu încearcă să interpreteze fiecare cuvânt din șir ca un parametru distinct din linia de comandă.

Caracterul asterisc (*) se potrivește cu orice șir de caractere, în timp ce semnul de întrebare (?) se potrivește cu un singur caracter. Să presupunem că în directorul curent se găsesc patru fișiere – `image1.pcx`, `image2.pcx`, `image3.pcx` și `image4.pcx`. Pentru a copia aceste fișiere în directorul `/mnt`, puteți utiliza comanda:

```
cp image?.pcx /mnt
```

Shellul Bash va înlocui semnul de întrebare cu orice alt caracter, copiind deci cele patru fișiere în directorul `/mnt`.

Al treilea format de înlocuire – `[...]` – substituie un anumit caracter dintr-un set specificat. De obicei, acest format se utilizează în combinație cu alte caractere de înlocuire pentru a restrânge setul fișierelor ale căror nume pot fi substituite. Pentru a vedea lista tuturor fișierelor din directorul `/etc/X11/xdm` care încep cu `x` sau `x`, tastați următoarea comandă:

```
ls /etc/X11/xdm/[xX]*
```



Observație

La expandarea formatului universal [...], dacă shellul Bash nu detectează nici un fișier al cărui nume satisface acest format, va lăsa specificarea universală intactă.

Un istoric al comenzilor utilizate

Pentru ca repetiția comenzilor lungi să se facă mai ușor, shellul Bash memorează până la 500 de comenzi executate anterior. În esență, Bash va păstra un *istoric al comenzilor* (o listă a comenzilor anterior executate). Pentru a vizualiza acest istoric, tastați comanda **history**. Shellul Bash va afișa o listă numerotată a comenzilor anterioare, inclusiv cele executate pe parcursul unor sesiuni anterioare. Această listă seamănă cu următoarea:

```
1 cd
2 ls -a
3 more /etc/X11/XF86Config
4 history
```

Dacă lista de comenzi se dovedește a fi prea lungă, puteți alege să vedeți numai ultimele câteva comenzi. De exemplu, pentru a vizualiza numai ultimele zece comenzi, tastați comanda:

```
history 10
```

Pentru repetarea unei comenzi din lista afișată de comanda **history**, este suficient să tastați un semn de exclamare (!) urmat de numărul acelei comenzi. Spre exemplu, pentru a repeta comanda cu numărul 3, tastați **!3**.

Puteți de asemenea să repetați o anumită comandă fără a-i cunoaște numărul. Să presupunem că ați tastat comanda **more /usr/lib/X11/xdm/xdm-config** într-un moment anterior și doriți să examinați din nou același fișier. Pentru a repeta comanda **more** precedentă, puteți utiliza comanda:

```
!more
```

În multe situații veți dori să repetați ultima comandă introdusă, eventual cu mici modificări. Spre exemplu, ați afișat conținutul directorului curent utilizând comanda **ls -l**. Pentru a repeta această comandă, tastați două semne de exclamare, după cum urmează:

```
!!
```

Uneori, veți dori să repetați comanda precedentă adăugându-i însă anumiți parametri suplimentari. Să presupunem că **ls -l** afișează prea multe fișiere. Puteți repeta această comandă, filtrând ieșirea sa cu ajutorul comenzii **more**, după cum urmează:

```
!! | more
```

Interpretorul Bash va înlocui cele două semne de exclamare cu comanda precedentă, adăugând apoi **| more** la sfârșitul comenzii.



Sugestie

O modalitate facilă de a executa din nou comenzi anterioare este apăsând tasta săgeata-sus, care determină interpretorul Bash să caute înapoi în lista de comenzi. Pentru a vă deplasa înainte în aceeași listă, apăsați tasta săgeată jos.

Editarea comenzilor

După ce regăsiți o anumită comandă, nu trebuie neapărat să o executați exact în forma în care este; aveți posibilitatea de a edita acea comandă. Shellul Bash suportă o varietate de comenzi de editare a liniei de comandă. Aceste comenzi sunt similare cu cele utilizate de editoarele **emacs** și **vi**.

Să presupunem că doriți să examinați fișierul **/etc/X11/XF86Config**, dar ați tastat din greșală comanda:

```
more /etc/X11/XF86config
```

După ce ați apăsător Enter, a apărut un mesaj de eroare care vă anunță că fișierul căutat nu există; vă dați astfel seama că litera **c** din numele **config** trebuia să fie mare. În loc de a tasta din nou integral linia de comandă, puteți utiliza următoarea comandă de editare pentru a rezolva problema:

```
^con^Con
```

Shellul Bash va interpreta această comandă înlocuind șirul de caractere **con** cu **Con** în comanda anterior executată.

În mod prestabilit, shellul Bash vă permite să editați linia de comandă utilizând un subset restrâns al comenzilor editorului **emacs**.

Dacă ați lucrat deja în **emacs**, puteți utiliza comenzile acestuia pentru a edita comenzile. De exemplu, pentru a regăsi comanda anterioară, puteți tasta **Ctrl-P**. Următoarele secvențe de taste vă permit apoi să editați linia de comandă:

Ctrl-B deplasare înapoi cu un caracter

Ctrl-F deplasare înainte cu un caracter

Ctrl-D șterge caracterul pe care se găsește cursorul

Pentru a insera un text, este suficient să tastați textul dorit. Chiar dacă editorul **emacs** dispune de un număr foarte mare de comenzi de editare, setul precedent prezentat vă este suficient pentru a edita comenzile Bash.

Pseudonime ale comenzilor

Atunci când ați configurat nucleul sistemului Linux și ați creat o dischetă de pornire, a fost necesar să vă deplasați de mai multe ori în directorul **/usr/src/linux/arch/i386/boot**. Într-o astfel de situație, se recomandă să creați o comandă rapidă, utilizând facilitatea de pseudonime (alias) ale comenzilor oferită de Bash:

```
alias goboot='cd /usr/src/linux/arch/i386/boot'
```

În mod intenționat, nu am utilizat nici o liniuță de subliniere și nici o literă mare în numele comenzii rapide, deoarece astfel acest nume poate fi tastat mai rapid (semnificația acestui nume fiind bine stabilită). După definirea acestui pseudonim, deplasarea în directorul menționat se realizează imediat, tastând la promptul interpretorului Bash:

```
goboot
```

După cum puteți observa, un pseudonim reprezintă un nume alternativ (de obicei mai scurt) pentru o comandă lungă. Interpretorul Bash înlocuiește pseudonimul cu definiția sa și execută comanda echivalentă.



Sugestie

Dacă trebuie să tastați frecvent o comandă mai lungă, se recomandă să definiți un pseudonim pentru acea comandă. Pentru a vă asigura că un pseudonim este disponibil în fiecare sesiune, includeți definiția sa în fișierul **.bash_profile** din directorul dumneavoastră personal.

Mulți utilizatori folosesc facilitatea de creare a pseudonimelor pentru a da nume familiare unor comenzi uzuale. Dacă utilizați în mod curent sistemul DOS și sunteți obișnuiți să listați conținutul unui director cu ajutorul comenzii **dir**, puteți pur și simplu să definiți **dir** ca pseudonim pentru comanda **ls** (comanda Linux care afișează conținutul unui director), după cum urmează:

alias dir=ls

Din acest moment, puteți tasta `dir` ori de câte ori doriți să afișați conținutul unui director.

O altă întrebare utilă a pseudonimelor este la redefinirea unor comenzi periculoase, cum este `rm`, astfel încât acestea să devină mai sigure. În mod prestabilit, comanda `rm` șterge unul sau mai multe fișiere precizate. Dacă din greșală ați tastat `rm *`, comanda va șterge toate fișierele din directorul curent. Spre exemplu, să presupunem că doriți să ștergeți toate fișierele cu extensia `.xwd`. (Acestea pot fi imagini reprezentând capturi ale ecranului, de care nu mai aveți nevoie.) Intenționați să introduceți comanda `rm *.xwd`, dar din greșală se strecoară un spațiu și comanda devine `rm * .xwd`. Ca rezultat, se va afișa mesajul:

```
rm: .xwd: No such file or directory
```

La început, veți fi stupefiat de acest mesaj, așa încât veți tasta `ls` pentru a vedea conținutul directorului încă o dată. Abia când veți observa că directorul nu mai conține nimic, veți realiza ce efect distructiv a avut spațiul strecurat din greșală între `*` și `.xwd`. Toate fișierele din director (inclusiv cele cu extensia `.xwd`) au fost între timp șterse definitiv.

Comanda `rm` oferă opțiunea `-i`, care vă cere o confirmare înainte de a șterge un fișier. Pentru ca această opțiune să fie utilizată în mod prestabilit, adăugați la fișierul `.bash_profile` din directorul personal definiția următorului pseudonim:

```
alias rm='rm -i'
```

Gata! De acum înainte, când veți utiliza comanda `rm` pentru a șterge fișiere, comanda va cere mai înainte o confirmare, ca în exemplul următor:

```
rm .bash_profile
rm: remove '.bash_profile'? n
```

Apăsați tasta `y` pentru a șterge fișierul; în caz contrar, tastați `n`.

Comenzi Linux

Interpretorul vă permite să rulați orice comandă Linux, dar trebuie să cunoașteți comenzile înainte de a le executa. Cum sistemul Linux reprezintă o variantă de UNIX, toate comenzile Linux sunt de fapt comenzi UNIX. Unele dintre cele mai importante comenzi sunt destinate lucrului cu sistemul de fișiere din Linux. Această secțiune prezintă aceste comenzi importante. Puteți încerca aceste comenzi de la promptul interpretorului în orice fereastră `xterm`.



Referință
încrucișată

În anexa B se găsește un ghid al comenzilor Linux, din care veți învăța despre un număr mult mai mare de comenzi.

Structura de directoare din Linux

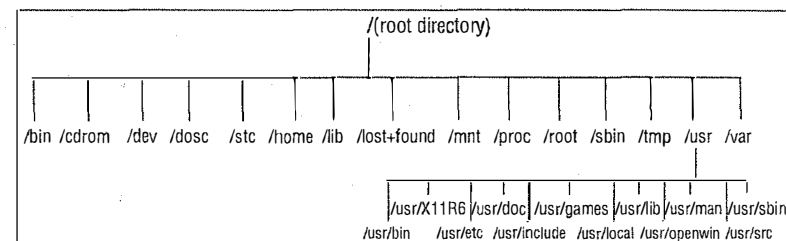
Ca orice alt sistem de operare, și sub Linux informațiile sunt organizate sub formă de fișiere. Fișierele sunt, la rândul lor, conținute în directoare. Un director poate avea subdirectoare, dând naștere unei structuri ierarhice. Spre deosebire de MS-DOS, în care apar unități individuale, sistemul de fișiere din Linux începe cu un director rădăcină; acesta este notat cu un singur caracter (`/`). Figura 6.1 prezintă structura de bază a sistemului de fișiere Linux, incluzând directorul rădăcină și câteva subdirectoare importante.

Multe dintre aceste directoare au scopuri bine precizate. Cunoscând scopul acestor directoare, vă va fi mai ușor să navigați prin sistemul de fișiere Linux. Un alt avantaj al cunoașterii rolului fiecărui director este că puteți intui cu ușurință unde să căutați anumite tipuri de fișiere atunci

când sunteți pus într-o situație inedită. Tabelul 6-2 descrie pe scurt rolul câtorva directoare principale din sistemul de fișiere Linux:

Figura 6-1

Structura
directoarelor
sistemului Linux



Tabelul 6-2 Directoarele sistemului Linux

Director	Descriere
<code>/</code>	Directorul rădăcină, care reprezintă baza sistemului de fișiere. Toate fișierele și directoarele sunt conținute logic în directorul rădăcină, indiferent de locațiile lor fizice.
<code>/bin</code>	Conține programele executabile care fac parte din sistemul de operare Linux. Multe comenzi Linux, cum sunt <code>cat</code> , <code>cp</code> , <code>ls</code> , <code>more</code> sau <code>tar</code> , se găsesc în <code>/bin</code> .
<code>/boot</code>	Conține nucleul sistemului Linux și alte fișiere necesare programului LILO, care încarcă sistemul de operare la pornire.
<code>/mnt/cdrom</code>	Director în care se montează de regulă unitatea CD-ROM.
<code>/dev</code>	Conține toate fișierele asociate dispozitivelor. Sistemul Linux tratează dispozitivele ca pe niște fișiere speciale, toate aceste fișiere găsindu-se în directorul <code>/dev</code> .
<code>/dosc</code>	Director în care de regulă este montată partiția DOS (în cazul în care sistemul dumneavoastră are o astfel de partiție).
<code>/etc</code>	Conține majoritatea fișierelor de configurare și a scripturilor de inițializare (acestea găsindu-se în subdirectorul <code>/etc/rc.d</code>).
<code>/home</code>	Locația convențională a directoarelor personale ale utilizatorilor. Spre exemplu, directorul personal al utilizatorului naba este <code>/home/naba</code> .
<code>/lib</code>	Conține biblioteci de funcții pentru C și alte limbaje de programare.
<code>/lost+found</code>	Director care păstrează fișierele pierdute. Fiecare partiție a unui disc are un director numit <code>lost+found</code> .
<code>/mnt</code>	Un director gol, utilizat de obicei pentru montarea temporară a unor dispozitive, ca de exemplu dischete sau partiții de disc. Conține subdirectoarele <code>/mnt/floppy</code> , în care se montează de regulă dischetele și <code>/mnt/cdrom</code> , utilizat pentru a monta unitățile CD-ROM (desigur, acestea pot fi montate și în alte directoare).
<code>/proc</code>	Director special care conține informații despre anumite aspecte legate de sistemul Linux.
<code>/root</code>	Directorul personal al utilizatorului root.
<code>/sbin</code>	Conține fișiere executabile reprezentând comenzi utilizate de regulă pentru activități de administrare a sistemului. În acest director se găsesc comenzi cum sunt <code>mount</code> , <code>halt</code> , <code>umount</code> sau <code>shutdown</code> .

(continuare)

Tabelul 6-2 Continuare

Director	Descriere
/tmp	Director temporar, care poate fi utilizat de orice utilizator pentru memorarea unor fișiere fără importanță (conținutul acestui director este considerat ca lipsit de importanță și ca atare este șters la fiecare repornire a sistemului).
/usr	Conține subdirectoarele utilizate de multe programe importante, printre care X Window System sau manualul online.
/var	Conține diferite fișiere utile pentru configurarea sistemului și directoare care păstrează informație volatilă.

Directorul /usr conține la rândul său o mulțime de subdirectoare utile. Tabelul 6-3 cuprinde o listă a celor mai importante subdirectoare care se găsesc în /usr.

Tabelul 6-3 Subdirectoare importante în /usr

Subdirectorul	Descriere
/usr/X11R6	Conține sistemul XFree86 (varianta pentru Linux a X Window System).
/usr/bin	Conține fișierele executabile pentru o mulțime de comenzi Linux, inclusiv pentru programe utilitare disponibile de obicei în Linux, dar care nu fac parte din esența sistemului de operare.
/usr/doc	Conține documentațiile pentru sistemul de operare Linux, ca și pentru multe programe utilitare ale sistemului (cum ar fi shellul Bash, <code>mttools</code> , programul de gestiune a fișierelor <code>xfm</code> sau programul de lucru cu imagini <code>xv</code>).
/usr/games	Conține o colecție de jocuri pentru Linux, cum ar fi <code>doom</code> , <code>dungeon</code> , <code>hangman</code> sau <code>snake</code> .
/usr/include	Conține fișierele antet (cu extensia <code>.h</code>) pentru dezvoltarea de programe în limbajele C și C++. Include de asemenea fișiere antet X11, în directorul <code>/usr/include/X11</code> .
/usr/lib	Conține bibliotecile pentru limbajele C și C++; de asemenea, conține biblioteci pentru X și Tcl.
/usr/local	Conține fișiere de importanță locală. Directorul <code>/usr/local/bin</code> conține, spre exemplu, programele executabile dezvoltate în sistemul dumneavoastră.
/usr/man	Conține manualul online (care poate fi citit utilizând comanda <code>man</code>).
/usr/sbin	Conține multe instrumente de administrare a sistemului, printre care comenzi pentru rularea și configurarea poștei electronice și a rețelei locale.
/usr/src	Conține codul sursă pentru nucleul sistemului Linux (esența sistemului de operare).

Navigarea prin directoare

În Linux, dacă intrați în sistem ca `root`, directorul dumneavoastră personal este `/root`. Pentru ceilalți utilizatori, directoarele personale se găsesc de regulă în directorul `/home`. Spre exemplu, directorul utilizatorului `naba` este `/home/naba`.

În mod prestabilit, un utilizator are permisiunea de a salva fișiere numai în directorul personal, dar poate crea un număr nelimitat de subdirectoare, pentru a asigura o mai bună organizare a fișierelor.

Toate interpretoarele Linux, printre care și Bash, implementează conceptul de director curent de lucru, care este directorul în care operează toate comenzile referitoare la fișiere și directoare. De exemplu, după intrarea în sistem, directorul curent de lucru este chiar directorul dumneavoastră personal. Pentru a determina directorul curent de lucru, utilizați comanda `pwd`. Pentru schimbarea directorului de lucru, utilizați comanda `cd`. Spre exemplu, pentru a schimba directorul curent în `/usr/doc`, tastați comanda:

```
cd /usr/doc
```

Mai departe, dacă doriți să vă deplasați în subdirectorul `bash-1.14.7` din `/usr/doc`, tastați comanda:

```
cd bash-1.14.7
```

În acest moment, dacă utilizați comanda `pwd`, aceasta va arăta că directorul curent de lucru este `/usr/doc/bash-1.14.7`. Prin urmare, vă puteți referi la numele unui director în două moduri:

- Prin intermediul unei căi absolute (cum este `/usr/doc`), care specifică în mod exact directorul în ierarhia de fișiere.

- Prin intermediul unui nume relativ (cum este `bash-1.14.7`, care reprezintă subdirectorul `bash-1.14.7` din directorul curent).

Dacă sunteți în directorul `/usr/doc` și tastați `cd bash-1.14.7`, directorul curent de lucru va deveni `/usr/doc/bash-1.14.7`; aceeași comandă, introdusă atunci când directorul curent este `/home/naba`, va încerca să schimbe directorul curent cu `/home/naba/bash-1.14.7`.



Sugestie

Utilizați comanda `cd` fără parametri pentru a reveni cu directorul curent în directorul dumneavoastră personal. De fapt, comanda `cd` fără parametri determină schimbarea directorului curent cu directorul listat în variabila de mediu `HOME`. Această variabilă conține însă în mod prestabilit directorul dumneavoastră personal.

Este util să observați că directorul specificat prin variabila `HOME` poate fi referit și prin intermediul caracterului `~` (tilda). Prin urmare, comanda `cd ~` determină schimbarea directorului curent cu directorul specificat de variabila `HOME`.



Văflor

Puteți utiliza o comandă rapidă pentru a vă referi la directorul personal al oricărui alt utilizator. Pentru a vă referi la directorul personal al unui utilizator, este suficient să prefixați numele utilizatorului cu caracterul special `~`. Prin urmare, `~naba` reprezintă directorul personal al utilizatorului `naba`, iar `~root` este directorul personal al utilizatorului `root`. Dacă doriți să vă deplasați în directorul personal al utilizatorului cu numele `emil`, este suficient să tastați `cd ~emil`.

Directoarele `.` și `..` au semnificații speciale. Un singur punct (`.`) indică directorul curent, în vreme ce două puncte (`..`) reprezintă părintele directorului curent. Spre exemplu, dacă directorul curent este `/usr/doc`, puteți schimba directorul curent cu `/usr` tastând simplu comanda:

```
cd ..
```

Afișarea fișierelor dintr-un director și permisiunile acordate acestora

Pe măsură ce vă deplasați prin directoarele din sistemul de fișiere Linux, veți dori să aflați ce se găsește în acele directoare. Conținutul unui director poate fi listat utilizând comanda `ls`. În mod prestabilit, comanda `ls` fără alte opțiuni afișează conținutul directorului curent într-un format compact pe patru coloane. Dacă ați intrat în sistem ca `root` și tastați `ls`, comanda `ls` va lista conținutul directorului `/root` în următoarea manieră:

Mail Xrootenv.0

În această listă, nu vă puteţi da seama dacă un anumit element reprezintă un fişier sau un alt director. Pentru a vedea detaliile complete ale conţinutului unui director, utilizaţi opţiunea -l a comenzii ls, după cum urmează:

ls -l

În directorul /root, comanda ls -l poate afişa spre exemplu:

```
total 2
drwx----- 2 root root 1024 Jan 26 00:32 Mail
-rw-r--r-- 1 root root 345 Apr 5 01:07 Xrootenv.0
```

Această listă oferă un volum considerabil de informaţii despre fiecare element dintr-un director, care poate fi un fişier sau un alt director. Examinând o linie începând din partea dreaptă, se observă că în prima coloană din dreapta se găseşte numele fişierului sau al subdirectorului. Data şi ora care apar înaintea numelui reprezintă data şi ora ultimei modificări a fişierului. Înainte de dată se găseşte dimensiunea fişierului, exprimată în baiţi.

La stânga coloanei care indică dimensiunea se găsesc grupul şi respectiv proprietarul fişierului. Următorul număr indică numărul legăturilor la fişierul respectiv. În fine, prima coloană din stânga prezintă permisiunile de acces, care determină cine poate citi, scrie sau executa fişierul.

Prima literă din stânga are o semnificaţie specială, după cum urmează:

- Dacă prima literă este l, fişierul reprezintă o legătură simbolică la un alt fişier.
- Dacă prima literă este d, atunci acel fişier este un director.
- Dacă prima literă este o liniuţă (-), atunci fişierul este unul obişnuit.

După prima literă, prima coloană din stânga conţine o secvenţă de nouă caractere, care apare sub forma rwxrwxrwx atunci când toate caracterele sunt prezente. Fiecare literă corespunde unei anumite permisiuni, înlocuirea ei printr-o liniuţă indicând absenţa permisiunii de a executa operaţia respectivă asupra fişierului. Este bine să consideraţi cele nouă litere ca fiind împărţite în trei grupuri de câte trei, interpretate după cum urmează:

■ Grupul din stânga determină permisiunile de citire, scriere şi execuţie ale proprietarului fişierului. Cu alte cuvinte, dacă în această poziţie observaţi rwx, proprietarul fişierului poate citi (r), scrie (w) şi executa (x) fişierul. Înlocuirea unei anumite litere printr-o liniuţă indică absenţa permisiunii respective. Prin urmare, şirul rw- înseamnă că proprietarul are drepturile de citire şi de scriere, dar nu şi pe cel de execuţie. De regulă, programele executabile (inclusiv programele de interpretor) au permisiune de execuţie.

■ Cele trei litere rwx din mijloc determină permisiunile de citire, scriere şi execuţie pentru orice utilizator aparţinând grupului din care face parte fişierul.

■ Grupul rwx din dreapta determină permisiunile de citire, scriere şi execuţie ale celorlalţi utilizatori (denumiţi colectiv prin sintagma: „restul lumii”).

În concluzie, un fişier cu permisiunile rwx este accesibil numai proprietarului său, în timp ce unul cu permisiunile rwxr- -r- - poate fi citit de oricine.

Utilizaţi comanda chmod pentru a modifica permisiunile unui fişier. Un program de interpretor (sau un script de interpretor) este de regulă un text pregătit cu ajutorul unui editor de texte. După ce salvaţi fişierul, trebuie mai întâi să îi daţi dreptul de execuţie pentru a-l putea rula şi testa. Pentru a da dreptul de execuţie unui script numit convert_images, tastaţi comanda:

chmod +x convert_images

Această comandă adaugă dreptul de execuţie pentru proprietar, grup şi restul lumii.

O caracteristică interesantă a comenzii ls este aceea că nu listează nici un fişier al cărui nume începe cu un punct. Pentru a vizualiza şi aceste fişiere, trebuie să utilizaţi comanda ls cu opţiunea -a, după cum urmează:

ls -a



Sugestie

Exemplul comenzii ls demonstrează o facilităţe interesantă a comenzilor din Linux. Majoritatea acestora au opţiuni cu lungimea de un caracter, precedate de semnul minus. (Acesta nu are aici semnificaţia din algebră, ci mai degrabă trebuie gândit ca o liniuţă de despărţire.) Atunci când doriţi să utilizaţi mai multe opţiuni, tastaţi o singură cratimă şi concatenaţi literele care reprezintă opţiunile una după alta. Aşadar, comanda ls -al este echivalentă cu ls -a -l.

Manipularea fişierelor

În multe cazuri, aveţi de copiat fişiere dintr-un director în altul. Pentru aceasta, utilizaţi comanda cp. Spre exemplu, pentru a copia fişierul

/usr/X11R6/lib/X11/fvwm2/system.fvwm2rc în alt director (cum ar fi directorul dumneavoastră personal), tastaţi următoarea comandă:

cp /usr/X11R6/lib/X11/fvwm2/system.fvwm2rc .fvwm2rc

Dacă doriţi să copiaţi un fişier în directorul curent, menţinându-i numele iniţial, utilizaţi un singur punct (.) ca al doilea parametru al comenzii cp. În consecinţă, comanda următoare copiază fişierul XF86Config din directorul /etc/X11 în directorul curent:

cp /etc/X11/XF86Config .

Comanda cp efectuează o copie a unui fişier, lăsând fişierul original intact.

O altă comandă a sistemului Linux, mv, mută un fişier existent într-o nouă locaţie. Copia originală dispare, în timp ce la destinaţie apare o nouă copie a fişierului. Comanda mv este utilizată şi pentru a schimba numele unui fişier. Spre exemplu, dacă doriţi să schimbaţi numele fişierului today.list în old.list, puteţi utiliza comanda mv după cum urmează:

mv today.list old.list

Pe de altă parte, dacă doriţi să deplasaţi fişierul today.list în subdirectorul saved, puteţi utiliza comanda:

mv today.list saved



O facilităţe interesantă oferită de mv este la deplasarea unor directoare întregi – împreună cu toate subdirectoarele şi fişierele pe care acestea le conţin – într-o nouă locaţie. Să presupunem că directorul numit data conţine o mulţime de fişiere şi subdirectoare; pentru a deplasa întreaga structură a directorului într-un alt director numit old_data, puteţi utiliza comanda:

mv data old_data

O altă operaţie frecventă este ştergerea fişierelor. Pentru a şterge fişiere, utilizaţi comanda rm. Spre exemplu, pentru a şterge fişierul old.list, tastaţi comanda:

rm old.list



Atenţie!

Fiiţi atenţi atunci când utilizaţi comanda rm, în special atunci când lucraţi ca root. Este foarte uşor să ştergeţi din greşeală fişiere extrem de importante utilizând această comandă. (Vezi secţiunea anterioară „Pseudonime ale comenzilor” pentru a învăţa cum puteţi evita distrugerea din neatenţie a unor fişiere importante cu comanda rm.)

În afară de copierea, redenumirea sau ștergerea fișierelor, o altă operație frecventă este vizualizarea conținutului acestora. Utilizați comanda `more` pentru a examina un fișier text pagină cu pagină. Pentru a vizualiza fișierul `/etc/X11/XF86Config`, utilizați comanda:

```
more /etc/X11/XF86Config
```

Comanda `more` se oprește după fiecare pagină; pentru deplasarea la pagina următoare, trebuie să apăsați tasta spațiu. Apăsați tasta `Enter` pentru a vă deplasa prin fișier linie cu linie.

O altă comandă utilă în Linux pentru vizualizarea fișierelor este `less`. (Numele acesteia este un joc de cuvinte, deoarece `less` are mai multe facilități decât `more`.¹) În timp ce `more` vă permite să vă deplasați în fișier doar înainte cu câte o pagină, utilizând comanda `less` vă puteți deplasa prin fișier atât înainte, cât și înapoi.

Lucrul cu directoare

Pentru organizarea fișierelor din directorul dumneavoastră personal, se recomandă să creați mai multe directoare noi. Pentru a crea un nou director, utilizați comanda `mkdir`. Spre exemplu, pentru a crea un subdirector numit `images` în directorul curent, tastați comanda:

```
mkdir images
```

După ce ați creat directorul, puteți utiliza comanda `cd images` pentru a vă deplasa în acest director.

Atunci când nu mai aveți nevoie de un anumit director, utilizați comanda `rmdir` pentru a-l șterge. Puteți șterge un director numai atunci când acesta este vid.

Căutarea fișierelor

Comanda `find` caută fișiere (și directoare) care satisfac anumite criterii de căutare. Comanda `find` este una dintre cele mai utile comenzi din sistemul Linux. Versiunea pentru Linux a acestei comenzi a fost realizată de organizația GNU și prezintă multe opțiuni suplimentare față de versiunea UNIX standard. Vom prezenta în continuare sintaxa comenzii standard, deoarece aceasta se poate utiliza atât sub Linux, cât și sub alte tipuri de sisteme UNIX.

Trebuie să admitem că la începutul activității în UNIX, orice utilizator poate fi surprins de această comandă. Majoritatea noilor utilizatori se rezumă doar la o singură formă a comenzii, aceea care permite căutarea în funcție de numele fișierului. Abia după un timp îndelungat, utilizatorii se aventurează să încerce forme ceva mai complicate ale lui `find`.

Să presupunem că dorim să găsim toate fișierele sau directoarele al căror nume începe cu `fvwm`. Pentru a efectua această operație de căutare, putem utiliza următoarea formă a comenzii `find`:

```
find / -name "fvwm*" -print
```

Această formă îi comunică lui `find` să înceapă căutarea din directorul rădăcină (`/`), să caute toate fișierele ale căror nume se potrivesc cu modelul `fvwm*` și să afișeze calea completă a fiecăruia dintre aceste fișiere.



Opțiunea `-print` și ghilimelele care delimitează șirul care trebuie căutat nu sunt necesare în Linux atât timp cât în directorul curent nu se găsesc fișiere al căror nume începe cu `fvwm`. Comanda prezentată anterior poate fi rescrisă mai simplu sub forma:

```
find / -name fvwm*
```

¹ În limba engleză, `less` – mai puțin, `more` – mai mult. (n. trad.)

Nu se întâmplă nimic dacă utilizați forma mai lungă a comenzii; oricum, aceasta este forma pe care trebuie să o utilizați în alte sisteme UNIX. Vom continua să prezentăm sintaxa comenzii `find` sub forma standard în care este utilizată în UNIX.

Puteți utiliza variante ale acestei forme simple a comenzii `find` pentru a căuta fișiere în orice director (inclusiv în subdirectoarele pe care acesta le conține). Dacă ați uitat în care dintre subdirectoarele din directorul personal ați pus fișierele numite `y2k*` (ale căror nume încep cu șirul `y2k`), le puteți căuta utilizând comanda:

```
find ~ -name "y2k*" -print
```

Atunci când veți stăpâni bine această formă a comenzii `find`, veți putea încerca să utilizați și celelalte opțiuni ale sale. Pentru a căuta numai anumite tipuri de fișiere (spre exemplu, numai directoare), utilizați opțiunea `-type`. Comanda următoare afișează toate directoarele situate pe primul nivel al ierarhiei sistemului de fișiere din Linux:

```
find / -type d -maxdepth 1 -print
```

Este puțin probabil să utilizați forme mai complicate ale comenzii `find` într-un sistem Linux obișnuit. Puteți studia oricând celelalte opțiuni ale comenzii `find` utilizând comanda:

```
man find
```

Scripturi de interpretor în Bash

Ideea de bază a sistemului UNIX a fost de a pune la dispoziția utilizatorului o colecție de comenzi simple, specializate, împreună cu toate facilitățile necesare pentru a conecta aceste comenzi. Conectarea a două comenzi se face în maniera uzuală, prin utilizarea ieșirii unei comenzi ca intrare pentru o alta. Bash, interpretorul prestabilit în Linux, oferă această facilități sub forma redirectării intrărilor și ieșirilor și a caracterelor de redirectare. Shell-ul Bash conține de asemenea facilități cum ar fi clauza `if`, care lansează comenzile numai dacă o anumită condiție este îndeplinită, sau clauza `for`, care repetă o secvență de comenzi de un număr specificat de ori. Puteți utiliza aceste facilități ale interpretorului Bash pentru a scrie programe numite *scripturi de interpretor*.

Scripturile de interpretor sunt foarte răspândite în lumea administratorilor de sistem. Dacă sunteți administratorul unui sistem, puteți construi o colecție proprie de scripturi de interpretor pentru a automatiza o parte dintre activitățile care se repetă frecvent. Spre exemplu, dacă discul se umple destul de des, se recomandă să căutați toate fișierele care depășesc o anumită dimensiune (să presupunem 1 MB) și la care nu s-a efectuat nici un acces în ultimele 30 de zile. În plus, este bine să trimiteți un mesaj prin e-mail către toți utilizatorii care au fișiere mari, solicitându-le să arhiveze sau să ștergă aceste fișiere. Toate aceste activități le puteți rezolva cu ajutorul unui singur script. Puteți începe cu următoarea comandă `find`, care identifică fișierele de dimensiuni mari:

```
find / -type f -atime +30 -size +1000k -exec ls -l {} \; > /tmp/largefiles
```

Această comandă creează fișierul `/tmp/largefiles`, care conține informații detaliate despre fișierele vechi care ocupă un spațiu prea mare. După ce ați obținut lista acestor fișiere, puteți utiliza în continuare alte câteva comenzi Linux – cum ar fi `sort`, `cut` sau `sed` – pentru a construi și apoi a trimite mesaje către utilizatorii care dețin fișiere mari, atrăgându-le atenția să le ștergă.

Următoarele câteva secțiuni prezintă o introducere în programarea în Bash.



Atenție! Atunci când doriți să încercați programe simple în interpretor, nu denumiți programele scrise cu numele test. Sistemul Linux include un program important numit test (/usr/bin/test). Acest program este utilizat în scripturile de interpretor pentru a testa anumite condiții, cum ar fi existența unui fișier sau existența permisiunii de citire asupra unui fișier. Dacă puneți unui script numele test, unele scripturi vor apela eronat scriptul dumneavoastră în locul programului test original.

Un script de interpretor simplu

Mai devreme în acest capitol, în secțiunea „Programe de shell”, ați observat că puteți include comenzi uzuale într-un fișier, după care, utilizând comanda chmod, puteți da fișierului creat permisiunea de execuție. În acest mod, ați creat un script de interpretor. După cum majoritatea comenzilor Linux acceptă opțiuni în linia de comandă, și scripturile Bash acceptă opțiuni în linia de comandă. În interiorul unui script, parametrii din linia de comandă pot fi referiți ca \$1, \$2 etc. Variabila specială \$0 se referă la numele scriptului.

Să considerăm următorul script Bash:

```
#!/bin/sh
echo "Numele scriptului este: $0"
echo Parametrul 1: $1
echo Parametrul 2: $2
```

Prima linie determină sistemul Linux să lanseze programul /bin/sh, care va prelucra ulterior restul liniilor scriptului. Numele /bin/sh corespunde interpretorului Bourne – primul interpretor disponibil în sistemele UNIX. În Linux, /bin/sh este o legătură simbolică la /bin/bash, programul executabil care reprezintă interpretorul Bash. Prin urmare, în Linux scripturile scrise pentru interpretorul Bourne sunt interpretate de către Bash (care este compatibil cu Bourne).

Dacă salvați acest script simplu într-un fișier numit simple și acordați permisiunea de execuție acelui fișier cu comanda chmod +x simple, veți putea apoi rula scriptul după cum urmează:

```
simple
Numele scriptului este ./simple
Parametrul 1:
Parametrul 2:
```

Mai multe amănunte despre programarea în shell

Chiar dacă aceste secțiuni reprezintă o introducere în domeniul programării în interpretor, domeniul este prea vast pentru a fi epuizat în acest capitol. Lucrarea *UNIX Power Tools*, scrisă în colaborare de câțiva specialiști de renume, constituie o bună sursă de documentare în ceea ce privește comenzile UNIX în general și programarea în shell în particular.

Numele scriptului de interpretor apare relativ la directorul curent de lucru, care este reprezentat printr-un punct. Cum scriptul a fost rulat fără parametri, nu va afișa nici un parametru.

Să încercăm să rulăm acum scriptul cu câțiva parametri, după cum urmează:

```
simple "Acesta este un parametru" al-doilea-parametru trei
Numele script-ului este ./simple
Parametrul 1: Acesta este un parametru
Parametrul 2: al-doilea-parametru
```

După cum puteți observa din acest exemplu, interpretorul consideră că un șir de caractere inclus între ghilimele reprezintă un singur parametru. Altfel, interpretorul va utiliza spațiile ca separatori între parametrii din linia de comandă.

Introducere în programarea în Bash

Ca orice alt limbaj de programare, Bash pune la dispoziție următoarele facilități:

- Variabile care pot memora valori, printre care se găsesc variabile special predefinite pentru accesul la parametrii în linia de comandă ai unui script interpretor și alte valori speciale
- Facilitatea de evaluare a expresiilor
- Structuri de control care vă permit să executați condiționat anumite comenzi sau să repetați o secvență de comenzi
- Facilitatea de a defini funcții care pot fi apelate în mai multe locuri pe parcursul unui script. Interpretorul Bash pune de asemenea la dispoziție multe funcții predefinite care pot fi utilizate în cadrul scripturilor.

Următoarele secțiuni prezintă câteva dintre facilitățile de programare oferite de Bash cu ajutorul unor exemple simple. Cum interpretorul pe care îl utilizați în mod prestabilit este Bash, puteți încerca aceste exemple tastându-le efectiv la prompt-ul de comandă într-o fereastră xterm.

Variabile

Variabilele se definesc în Bash similar cu variabilele de mediu. În consecință, puteți defini o variabilă în modul următor:

```
count=12 # observati ca nu apar spatii
```

Pentru a utiliza valoarea unei variabile, prefixați numele variabilei cu semnul dolar (\$). \$PATH, spre exemplu, reprezintă valoarea variabilei PATH (este vorba într-adevăr de binecunoscuta variabilă de mediu PATH). Pentru a afișa valoarea variabilei numite count, trebuie să utilizați următoarea comandă:

```
echo $count
```

Bash utilizează anumite variabile speciale pentru accesul la parametrii din linia de comandă. Într-un script de interpretor, \$0 se referă la numele scriptului. Variabilele \$1, \$2 etc. reprezintă parametrii din linia de comandă. Variabila \$* memorează întreg șirul parametrilor din linia de comandă sub forma unei singure variabile, în timp ce \$? conține codul de terminare al ultimei comenzi executate de către interpretor.

Este posibil ca în scripturile de interpretor să cereți utilizatorului să introducă o anumită valoare, utilizând apoi comanda read pentru a memora valoarea introdusă într-o variabilă. Iată un exemplu:

```
echo -n "Introduceți valoarea: "
read value
echo "Ați introdus: $value"
```



Sugestie

Opțiunea -n determină comanda echo să nu mai adauge automat un caracter de trecere la o linie nouă după afișarea unui șir de caractere.

Structuri de control

În scripturile Bash, structurile de control – ca de exemplu if, case, for sau while – depind de codul de terminare al unei comenzi pentru a decide pasul care va fi efectuat în continuare.

Fiecare comandă returnează, după ce își termină execuția, un *cod de terminare*. Acesta este un număr care oferă indicații asupra terminării cu succes a comenzii. Prin convenție, o comandă

terminată cu succes returnează un cod de terminare egal cu zero. (Rețineți corect această regulă: zero indică terminarea cu succes.) Dacă în rularea comenzii a survenit o eroare, aceasta returnează un cod de terminare nenul.

Înainte de a edita un anumit fișier cu editorul `vi`, se recomandă să efectuați o copie a celui fișier. O modalitate de a efectua copia unui fișier este următoarea:

```
#!/bin/sh
if cp "$1" "$1"
then
    vi "$1"
else
    echo "Nu am putut crea copia de siguranta"
fi
```

Scriptul ilustrează sintaxa structurii decizionale `if-then-else` și arată totodată cum structura `if`, pentru a decide acțiunea pe care o va efectua la pasul următor, utilizează codul de terminare al comenzii `cp`. Dacă acesta este zero, scriptul invocă editorul `vi` pentru a edita fișierul; în caz contrar, afișează un mesaj și își încheie execuția.



Nu uitați particula finală `fi` care marchează terminarea structurii `if`. Omiterea acesteia este una din sursele frecvente de erori în scripturile Bash.

Sugestie

Shellul Bash include comanda `test`, care permite evaluarea oricărei expresii, utilizând valoarea acelei expresii în locul codului de terminare a unei comenzi. Să presupunem că doriți să scrieți un script care permite editarea unui fișier existent. Utilizând comanda `test`, puteți scrie un astfel de script în modul următor:

```
#!/bin/sh
if test -f "$1"
then
    vi "$1"
else
    echo "Fișier inexistent"
fi
```

Forma mai scurtă a comenzii `test` omite cuvântul cheie `test`, introducând opțiunile comenzii între paranteze pătrate. Utilizând această notație, scriptul precedent se rescrie astfel:

```
#!/bin/sh
if [ -f "$1" ]
then
    vi "$1"
else
    echo "Fișier inexistent"
fi
```

O altă structură de control frecvent utilizată este bucla `for`. Scriptul care urmează calculează suma numerelor de la 1 la 10:

```
#!/bin/sh
sum=0
for i in 1 2 3 4 5 6 7 8 9 10
do
    sum="expr $sum + $i"
done
echo "Suma = $sum"
```

Acest exemplu ilustrează de asemenea modul de utilizare a comenzii `expr` pentru evaluarea unei expresii.

Funcții predefinite în Bash

Interpretorul Bash dispune de mai mult de 50 de funcții predefinite, de la unele extrem de comune cum sunt `cd` sau `pwd`, până la altele mai puțin utilizate. Aceste funcții predefinite pot fi utilizate în orice script Bash. Volumul lucrării de față nu ne permite să prezentăm în detaliu toate aceste funcții predefinite. Puteți afla mai multe despre aceste funcții consultând manualul online pentru interpretorul Bash, utilizând comanda `man bash`.

Utilizarea limbajului Perl ca limbaj de comenzi

În mod oficial, abrevierea *Perl* provine de la *Practical Extraction Report Language*. Limbajul Perl a fost creat de Larry Wall pentru extragerea informațiilor din fișierele text și pentru pregătirea acestor informații în vederea alcătuirii unor rapoarte. Programele scrise în limbajul Perl sunt interpretate și apoi executate de programul numit `perl`, care reprezintă interpretorul limbajului. CD-ROM-ul atașat acestei lucrări include programul `perl`; acesta este de regulă instalat în sistemul dumneavoastră în directorul `/usr/bin`.

Limbajul Perl este disponibil pentru o mare varietate de platforme deoarece, întocmai ca și sistemul Linux, este distribuit gratuit. De asemenea, Perl este foarte răspândit în rândul utilizatorilor și administratorilor de sistem ca limbaj utilizat pentru realizarea scripturilor, motiv pentru care am dedicat această secțiune unei prezentări generale a limbajului Perl, subliniindu-i facilitățile esențiale. În capitolul 8, veți învăța un alt limbaj utilizat la realizarea de scripturi (Tcl/Tk), care oferă de asemenea posibilitatea creării unei interfețe grafice pentru un anumit script.

După cum deja știți, termenul de *script* este sinonim cu cel de *program*. Spre deosebire de programele scrise în C sau C++, programele Perl nu trebuie compilate; programul `perl` realizează direct interpretarea și lansarea în execuție a programelor Perl. Termenul de *script* este îndeosebi întrebuințat pentru a denumi programele interpretate, scrise în limbajul de programare al unui interpretor sau în Perl. (Pentru a fi mai exacti, programul `perl` nu interpretează imediat un program Perl, ci îl convertește mai întâi într-un format intermediar pe care ulterior îl execută.)



Observație

Dacă aveți experiență în programarea în interpretor sau în limbajul C, puteți învăța repede Perl. Dacă însă nu ați scris niciodată un program, nu puteți deveni peste noapte un programator experimentat în Perl. Vă recomandăm să începeți cu un subset minimal de facilități oferite de Perl, ignorându-le pe cele pe care nu le înțelegeți. Veți putea apoi să adăugați treptat noi facilități ale limbajului la volumul dumneavoastră de cunoștințe.

Este Perl instalat în sistem?

Înainte de a trece la prezentarea elementelor de bază ale limbajului Perl, este necesar să verificați dacă programul `perl` este instalat în sistemul dumneavoastră. Tastați următoarea comandă:

```
which perl
```

Comanda `which` vă spune dacă poate găsi programul precizat în directoarele care alcătuiesc variabila de mediu `PATH`. Dacă interpretorul `perl` este instalat, comanda va afișa următorul rezultat:

```
/usr/bin/perl
```

Dacă însă comanda `which` vă spune că fișierul căutat nu există în directoarele din calea curentă, nu înseamnă în mod necesar că programul `perl` nu este instalat. Este posibil ca directorul `/usr/bin` să nu se găsească în variabila de mediu `PATH`. Pentru a verifica dacă directorul `/usr/bin` se găsește în calea curentă, aveți două posibilități: fie tastați `echo $PATH`, fie

examinați mesajul de eroare afișat de comanda `which` (acesta include directoarele din variabila `PATH`). Dacă directorul `/usr/bin` nu se găsește în variabila `PATH`, tastați următoarea comandă pentru a modifica această variabilă:

```
PATH=$PATH:/usr/bin; export PATH
```

Încercați acum din nou comanda `which perl`. Dacă și acum vi se semnalează o eroare, interpretorul `perl` nu este instalat în sistem. Pentru a instala interpretorul `perl` de pe CD-ROM-ul anexat acestei cărți, trebuie să parcurgeți următoarele etape:

1. Intrați în sistem ca `root`.
2. Introduceți CD-ROM-ul anexat acestei cărți în unitatea CD-ROM și montați-l utilizând următoarea comandă:

```
mount /dev/hdc /mnt/cdrom
```

3. Tastați următoarea comandă pentru a vă deplasa în directorul în care se găsesc pachetele sistemului Red Hat:

```
cd /mnt/cdrom/RedHat/RPMS
```

4. Tastați următoarea comandă `rpm` (Red Hat Package Manager) pentru a instala Perl:

```
rpm -i perl*
```

Dacă interpretorul `perl` este instalat în sistem, tastați următoarea comandă pentru a determina numărul versiunii instalate:

```
perl -v
```

Iată ce afișează de obicei această comandă:

```
This is perl, version 5.004_01
```

```
Copyright 1987-1997, Larry Wall
```

Perl may be copied only under the terms of either the Artistic License or the GNU General Public License, which may be found in the Perl 4.0 source kit.

Acest mesaj vă aduce la cunoștință că versiunea instalată este 5.004, nivelul de patch este 01, iar drepturile de autor aparțin creatorului limbajului Perl, Larry Wall. Limbajul Perl este însă distribuit gratuit în termenii acordului GNU General Public License.

Pe CD-ROM-ul anexat acestei cărți se găsește versiunea 5.004_01 a limbajului Perl. Puteți obține cea mai recentă versiune a limbajului prin World Wide Web, în situl Comprehensive Perl Archive Network (CPAN). Următoarea adresă vă va conecta imediat la situl CPAN cel mai apropiat de dumneavoastră:

```
http://www.perl.com/CPAN/
```

Primul dumneavoastră script în Perl

Limbajul Perl dispune de multe facilități moștenite din C. După cum bine știți, majoritatea cârților de C încep cu un program simplu care afișează mesajul `Hello, World!` pe terminalul dumneavoastră. Cum Perl este un limbaj interpretat, puteți realiza această operație direct din linia de comandă. Dacă veți introduce:

```
perl -e 'print "Hello, World!\n";'
```

sistemul va afișa mesajul:

```
Hello, World!
```

Această comandă utilizează opțiunea `-e` a programului `perl` pentru a transmite programului Perl un parametru în linia de comandă pentru interpretorul de Perl. În acest caz, programul Perl constă dintr-o singură linie:

```
print "Hello, World!\n";
```

Pentru a transforma această linie într-un script, este suficient să o salvați într-un fișier separat, în a cărui primă linie să includeți o directivă care va lansa interpretorul `perl` (ca și în cazul scripturilor de interpretor, în care în prima linie se scrie directiva `#!/bin/sh`, în scopul utilizării interpretorului Bourne pentru a le prelucra).

Pentru a încerca să rulați un script în Perl, trebuie să parcurgeți următoarele etape:

1. Utilizați un editor de texte, cum sunt `vi` sau `emacs`, pentru a salva următoarele linii într-un fișier numit `hello`:

```
#!/usr/bin/perl
# Acesta este un comentariu.
print "Hello, World!\n";
```

2. Dați fișierului `hello` drept de execuție, utilizând comanda:

```
chmod +x hello
```

3. Rulați scriptul Perl introducând la promptul interpretorului:

```
hello
Hello, World!
```



Secret

Observați că prima linie a unui script Perl începe cu secvența de caractere `#!`, urmată de calea completă către programul `perl`. Atunci când prima linie a unui script începe cu `#!`, interpretorul elimină această secvență, adaugă numele scriptului la sfârșit și apoi execută script-ul. Prin urmare, dacă scriptul se numește `hello`, iar prima linie este `#!/usr/bin/perl`, interpretorul va executa comanda:

```
/usr/bin/perl hello
```

Mai multe amănunte despre Perl

Câteva secțiuni din capitolul de față sunt dedicate introducerii în limbajul Perl și prezentării unor exemple simple. Această discuție este însă departe de a epuiza subiectul. Dacă doriți să aprofundați acest limbaj, puteți consulta una dintre următoarele lucrări:

Larry Wall, Tom Christiansen și Randal L. Schwartz – *Programming Perl, 2nd Edition* (Editura O'Reilly & Associates, 1996)

Randal L. Schwartz – *Learning Perl* (Editura O'Reilly & Associates, 1993)

Naba Barkakati – *Discover Perl 5* (IDG Books Worldwide, 1997)

Lucrarea de bază dedicată limbajului Perl este chiar cea scrisă de creatorul limbajului, Larry Wall (chiar dacă nu se adresează începătorilor). Lucrarea lui Randal Schwartz este mai mult orientată pe învățarea programării în Perl. Cartea scrisă de autorul lucrării de față prezintă limbajul Perl pas cu pas, fiecare secțiune fiind însoțită de programe scurte, care evidențiază noțiunile esențiale ale limbajului.

O prezentare scurtă a limbajului Perl

Majoritatea limbajelor de programare, printre care și Perl, au anumite caracteristici comune:

- Datele de diferite tipuri sunt memorate cu ajutorul *variabilelor*. Acestea pot fi gândite ca niște locuri în care sunt păstrate datele (similare cutiilor poștale), fiind caracterizate printr-un nume și un spațiu în care pot memora datele. Conținutul unei variabile este chiar valoarea acesteia.

- *Expresiile* pot combina variabilele utilizând *operatori*. O expresie poate, de exemplu, să adune mai multe variabile; o alta poate extrage un anumit subsir dintr-un şir de caractere.
- *Instrucţiunile* pot executa anumite acţiuni, cum ar fi atribuirea unei valori pentru o variabilă sau afişarea unui şir de caractere.
- *Instrucţiunile care controlează fluxul programului*. Acestea permit execuţia instrucţiunilor din program într-o ordine diferită în funcţie de valoarea unei anumite expresii. Setul de instrucţiuni de control utilizat în majoritatea limbajelor moderne cuprinde instrucţiunile `for`, `do-while`, `while` şi `if-then-else`.
- *Funcţiile* (denumite şi *subrutine* sau *rutine*) permit gruparea mai multor instrucţiuni sub un nume. Prezenţa funcţiilor permite execuţia unui set de instrucţiuni prin apelul funcţiei care cuprinde acele instrucţiuni. În mod uzual, limbajele de programare pun la dispoziţie câte un set de funcţii predefinite.

În următoarele câteva secţiuni sunt prezentate pe scurt facilităţile esenţiale ale limbajului Perl, acestea fiind ilustrate cu ajutorul unor exemple simple.

Sintaxa de bază a limbajului Perl

Ca şi în C, nici în Perl nu există restricţii de amplasare a cuvintelor cheie în cadrul unui program. Programele Perl sunt de regulă memorate în fişiere cu extensia `.pl`, dar aceasta nu este o regulă obligatorie.

Fiecare instrucţiune se termină în Perl, ca şi în C, cu caracterul „;” (punct şi virgulă). Începutul unui comentariu este marcat prin semnul (`#`). Interpretorul `perl` va ignora conţinutul unei linii care începe cu semnul `#`.

Instrucţiunile compuse sunt incluse între acolade (`{...}`). Această convenţie este de asemenea moştenită din C.

Variabile

Variabilele în Perl nu trebuie declarate înainte de a fi folosite, aşa cum se procedează în C. Variabilele pot fi recunoscute cu uşurinţă în cadrul scripturilor Perl, deoarece numele acestora încep cu caractere speciale: semnul „at” (`@`), semnul dolar (`$`) sau semnul procent (`%`). Fiecare dintre aceste caractere speciale corespunde unui anumit tip de variabile. Variabilele în Perl pot fi de trei tipuri:

- *Variabile scalare*, care reprezintă tipurile elementare de date: întregi, numere în virgulă mobilă şi şiruri de caractere. Variabilele scalare sunt precedate de semnul dolar (`$`). Iată câteva exemple:

```
$maxlines = 256;
$title = "Secretele sistemului Linux, editia a doua";
```

- *Variabilele de tip masiv* sunt colecţii de variabile scalare. Orice variabilă de tip masiv este prefixată cu simbolul „@”. În consecinţă, următoarele declaraţii reprezintă masive:

```
@pages = (62, 26, 22, 24);
@commands = {"start", "stop", "draw", "exit"};
```

- *Masivele (tablourile) asociative* sunt colecţii de perechi cheie-valoare, în care fiecare cheie este un şir de caractere, iar valoarea poate fi orice tip de variabilă scalară. Tablourile asociative sunt prefixate cu semnul procentual (`%`). Tablourile asociative se utilizează pentru a asocia un nume cu o anumită valoare. Puteţi de exemplu memora cantitatea de spaţiu pe disc ocupată de fiecare utilizator într-un tablou asociativ, după cum urmează:

```
%disk_usage = ("root", 147178, "naba", 28547, "emily",
55, "ivy", 60);
```

Din cauză că fiecare tip de variabilă este prefixat cu un caracter diferit, puteţi utiliza acelaşi nume pentru variabile de tipuri diferite. Prin urmare, într-un acelaşi program Perl pot apărea variabilele diferite `%disk_usage`, `@disk_usage` şi `$disk_usage`.

Variabile scalare

Variabilele scalare reprezintă tipul esenţial de date în Perl. Numele fiecărei variabile scalare este prefixat cu semnul dolar (`$`). În cele mai multe cazuri, o variabilă scalară apare pentru prima dată într-o instrucţiune de atribuire care o iniţializează. Puteţi utiliza o variabilă fără a o iniţializa. Valoarea prestabilită a numerelor este zero, iar valoarea prestabilită a unui şir de caractere este şirul vid. Pentru a vedea dacă un scalar este definit, utilizaţi funcţia `defined`, după cum urmează:

```
print "Nume nedefinit!\n" if ! (defined $name);
```

Expresia `(defined $name)` are valoarea 1 dacă variabila `$name` este definită. Puteţi anula definiţia unei variabile utilizând funcţia `undef`. Spre exemplu, puteţi anula definiţia variabilei `$name`, utilizând instrucţiunea:

```
undef $name;
```

Variabilele sunt evaluate în Perl în conformitate cu contextul. Iată în continuare un script care iniţializează şi tipăreşte câteva variabile:

```
#!/usr/bin/perl
$title = "Secrete Linux, editia a doua";
$count1 = 550;
$count2 = 375;
```

```
$total = $count1 + $count2;
```

```
print "Titlul: $title -- $total pagini\n";
```

Atunci când rulaţi acest program Perl, vor fi afişate următoarele rezultate:

```
Titlul: Secrete Linux, editia a doua -- 925 pagini
```

După cum observaţi din aceste instrucţiuni, atunci când două valori numerice sunt adunate, se utilizează reprezentările lor numerice; atunci când variabila `$total` este afişată, se utilizează reprezentarea ei sub formă de şir de caractere.

Un alt aspect interesant al limbajului Perl este că evaluează toate variabilele dintr-un şir de caractere cuprins între ghilimele („...”). Pe de altă parte, scriind un şir de caractere între apostrofuri simple („...”), Perl va lăsa acest şir nemodificat. Dacă în programul de mai sus aţi fi scris:

```
print 'Titlul: $title -- $total pagini\n';
```

cu apostrofuri în loc de ghilimele, interpretorul ar fi afişat următoarele rezultate:

```
Titlul: $title -- $total pagini\n
```

fără să treacă la linia nouă.



O variabilă extrem de utilă în Perl este `$_` (semnul dolar, urmat de liniuţa de subliniere). Această variabilă este denumită variabila prestabilită. Interpretorul de Perl stabileşte valoarea acestei variabile în funcţie de context. Atunci când interpretorul citeşte valori de la intrarea standard, `$_` memorează linia curent citită. Când interpretorul caută un anumit model, `$_` va conţine modelul prestabilit de căutare.

Tablouri

Un *tablou* este o colecție de valori scalare. Numele unui tablou începe cu caracterul „at” (@). Ca și în C, tablourile sunt indexate începând cu zero. Puteți avea acces la elementele dintr-un tablou utilizând un index. Limbajul Perl alocă spațiu în memorie pentru tablouri în mod dinamic.

Să considerăm următorul script simplu:

```
#!/usr/bin/perl
@commands = ("start", "stop", "draw", "exit");

$numcmd = @commands;
print "Tabloul contine $numcmd comenzi. Prima comanda este:
      $commands[0]\n";
```

Atunci când veți rula acest script, va afișa următorul rezultat:

Tabloul contine 4 comenzi. Prima comanda este: start

După cum puteți observa, dacă atribuim unui scalar un tablou, scalarul va primi valoarea numărului de elemente din tablou. Primul element al tabloului @commands este referit prin expresia \$commands[0], deoarece indexarea începe de la zero. În consecință, al patrulea element din tablou este \$commands[3].

Relativ la un anumit tablou, se pot defini două variabile scalare speciale. Variabila \$[este indexul curent de bază, care în mod prestabilit are valoarea zero. Scalarul \$#nume_tablou (unde nume_tablou este numele unei variabile de tip tablou) are ca valoare indexul ultimului element din tablou. În consecință, pentru tabloul @commands, \$#commands are valoarea 3.

Puteți tipări un întreg tablou cu o singură instrucțiune print, după cum urmează:

```
print "@commands\n";
```

În urma execuției acestei instrucțiuni, interpretorul perl va afișa următorul rezultat:

```
start stop draw exit
```

Tablouri asociative

Variabilele de tipul tablou asociativ, care sunt prefixate cu semnul procentual (%), reprezintă facilități caracteristice limbajului Perl. Utilizând tablouri asociative, puteți indexa un tablou utilizând un șir de caractere cu rol de index. Un exemplu bun de tablouri asociative este tabloul %ENV, definit automat în Perl. În Perl, %ENV reprezintă tabloul variabilelor de mediu, la care puteți avea acces utilizând numele unei anumite variabile ca index. Următoarea instrucțiune Perl afișează valoarea curentă a variabilei de mediu PATH:

```
print "PATH = $ENV{PATH}\n";
```

În urma execuției acestei instrucțiuni, interpretorul Perl va afișa valoarea curentă a variabilei PATH. Spre deosebire de tablourile obișnuite, trebuie să utilizați acolade pentru a indexa un element dintr-un tablou asociativ.

Limbajul Perl dispune de multe funcții predefinite, cum sunt delete, each, keys și values, care vă permit să lucrați cu tablouri asociative.

Operatori și expresii

Operatorii sunt utilizați pentru a combina și compara variabilele în Perl. Operatorii matematici uzuali sunt cei de adunare (+), scădere (-), înmulțire (*) și împărțire (/). Limbajul Perl dispune de un set de operatori aproape identic cu cel din C. Ca urmare a utilizării operatorilor pentru a combina mai multe variabile, rezultă expresiile. Fiecare expresie are o anumită valoare.

Variable predefinite în Perl

În Perl există câteva variabile predefinite care conțin informații utile de care aveți nevoie într-un script Perl. Iată câteva dintre cele mai importante variabile predefinite în Perl:

@ARGV este șirul de caractere care conține opțiunile din linia de comandă ale unui script. Prima opțiune este \$ARGV[0], cea de-a doua \$ARGV[1] etc.

%ENV este un tablou asociativ care conține variabilele de mediu. Puteți utiliza acest tablou utilizând numele variabilelor căutate pe post de chei. Prin urmare, \$ENV{HOME} reprezintă directorul dumneavoastră personal, iar \$ENV{PATH} este calea pe care interpretorul o utilizează pentru a căuta comenzi.

\$\$_ este identificatorul de proces al scriptului curent.

\$< este identificatorul utilizatorului care rulează scriptul.

\$? este starea returnată de ultimul apel system.

\$_ este parametrul prestabilit utilizat de multe alte funcții.

\$0 este numele scriptului curent.

Iată câteva exemple de expresii în Perl:

```
$error < 0
$count == 10
$count + $i
$users[$i]
```

Aceste expresii utilizează operatorii de comparație (primele două), operatorii aritmetici (cea de-a treia) și operatorul de indexare într-un tablou (ultima).



Atenție!

Nu utilizați în Perl operatorul == pentru a testa dacă două șiruri de caractere coincid; operatorul == funcționează corect numai dacă este aplicat numerelor. Pentru a testa egalitatea șirurilor de caractere în Perl, se folosește operatorul eq, a cărui denumire provine din FORTRAN. Puteți utiliza eq pentru a testa dacă două șiruri de caractere coincid, după cum urmează:

```
if ($input eq "stop") {exit;}
```

Alți operatori de comparație pentru șirurile de caractere cu nume provenind din FORTRAN sunt ne (inegalitate), lt (mai mic), gt (mai mare), le (mai mic sau egal) și ge (mai mare sau egal). Puteți utiliza de asemenea operatorul cmp pentru a compara două șiruri de caractere. Valoarea returnată de acesta este -1, 0 sau 1, după cum primul șir se găsește în ordinea lexicală înaintea celui de-al doilea, coincide cu acesta sau respectiv se află după al doilea.

În limbajul Perl există de asemenea câțiva operatori inexistenți în C. Unul dintre aceștia este operatorul de exponențiere, care în C lipsește, deși în FORTRAN era prezent. Limbajul Perl utilizează simbolul ** ca operator de ridicare la putere. Puteți scrie deci un cod de forma:

```
$x = 2;
$y = 3;
$z = $x ** $y # z va avea valoarea 8 (2 ridicat la puterea 3)
$y **= 2; # y va avea acum valoarea 9 (3 ridicat la puterea 2)
```

Puteți inițializa un șir de caractere cu șirul vid utilizând operatorul de listă vidă () – ca în exemplul următor:

```
@commands = ();
```

Operatorul punct (.) vă permite să concatenați două șiruri, după cum urmează:

```
$part1 = "Hello, ";
$part2 = "World!";
$message = $part1.$part2; # $message va fi "Hello, World!"
```

Un operator ciudat, dar destul de util, este *operatorul de repetiție*, notat cu `x=`. Puteți utiliza operatorul `x=` pentru a repeta un șir de caractere de un număr precizat de ori. Să presupunem că doriți să inițializați un șir de caractere cu 65 de asteriscuri (*). Următorul exemplu vă arată cum puteți realiza aceasta utilizând operatorul `x=`:

```
$marker = "";  
$marker x= 65; # $marker este acum un sir de 65 de asteriscuri
```

Un alt operator puternic al limbajului Perl este *operatorul de delimitare a unui domeniu*, specificat prin două puncte (..). Puteți inițializa un tablou cu ușurință utilizând acest operator. Iată câteva exemple:

```
@numerals = (0..9); # @numerals contine numerele din lista 0,1,2,3,4,5,6,7,8,9  
@alphabet = ('A'..'Z'); # @alphabet este lista literelor mari de la A la Z.
```

Expresii regulate

Dacă nu sunteți chiar un novice în UNIX, probabil cunoașteți comanda `grep`, care vă permite să căutați un anumit șir de caractere într-o colecție de fișiere. În continuare, iată o formă a comenzii `grep` care caută toate aparițiile șirurilor de caractere `blaster` sau `Blaster` în toate fișierele ale căror nume se termină cu `.c`:

```
cd /usr/src/linux/drivers/cdrom  
grep "[bB]laster" *.c
```

Iată, de exemplu, rezultatele pe care această comandă le poate afișa într-un sistem:

```
cdu31a.c: { 0x230, 0 } /* SoundBlaster 16 card */  
sbpcd.c: * Works with SoundBlaster compatible cards and with "no-sound"  
sbpcd.c: 0x230, 1, /* SoundBlaster Pro and 16 (default) */  
sbpcd.c: 0x250, 1, /* OmnicD default, SoundBlaster Pro and 16 */  
sbpcd.c: 0x270, 1, /* SoundBlaster 16 */  
sbpcd.c: 0x290, 1, /* SoundBlaster 16 */  
sbpcd.c: static const char *str_sb = "SoundBlaster";  
sbpcd.c: static const char *str_sb_l = "soundblaster";  
sbpcd.c: * sbpcd=0x230,SoundBlaster  
sbpcd.c: msg(DBG_INF, " LIL0 boot:...  
sbpcd=0x230,SoundBlaster\n");  
sjcd.c: * the SoundBlaster/Panasonic style CDR0M interface. But today, the
```

După cum observați, această comandă `grep` determină toate aparițiile șirurilor `blaster` și `Blaster` în fișierele cu extensia `.c`.

Parametrul `"[bB]laster"` al comenzii `grep` este cunoscut sub numele de *expresie regulată*. Expresiile regulate sunt modele care se potrivesc cu anumite șiruri de caractere. Construcția expresiilor regulate se face utilizând un set de operatori și un set de reguli similare cu cele care se aplică pentru expresiile aritmetice. De exemplu, o listă de caractere cuprinse între paranteze pătrate (`[...]`) corespunde cu fiecare dintre caracterele din listă. Prin urmare, expresia regulată `"[bB]laster"` corespunde unui set de două șiruri de caractere, anume:

```
blaster Blaster
```

Limbajul Perl suportă aceleași expresii regulate care sunt cunoscute și de către comanda `grep`. Multe alte programe UNIX, ca de exemplu editoarele `vi` și `sed` (stream editor), lucrează cu expresii regulate. Scopul unei expresii regulate este acela de a căuta un anumit model de șiruri de caractere într-un fișier. Acesta este motivul pentru care editoarele din UNIX implementează mecanismul expresiilor regulate.

Limbajul Perl vă permite să construiți expresii regulate complexe, utilizând însă un set de reguli relativ simple. În esență, o expresie regulată reprezintă o secvență de caractere, unele

dintre acestea având semnificații speciale. Principalele reguli de interpretare a caracterelor sunt prezentate în tabelul 6-4.

Tabelul 6-4 Reguli de interpretare a caracterelor din expresiile regulate

Expresie	Semnificație
.	Orice caracter, exceptând caracterul „linie nouă”
x*	Zero sau mai multe apariții ale caracterului x
x+	Una sau mai multe apariții ale caracterului x
x?	Zero sau o apariție a caracterului x
[...]	Oricare dintre caracterele cuprinse în paranteze
x{n}	Exact n apariții ale caracterului x
x{n,}	Cel puțin n apariții ale caracterului x
x{,m}	Zero sau cel mult m apariții ale caracterului x
x{n,m}	Un număr de apariții ale caracterului x cuprins între n și m
\$	Sfârșitul unei linii
\0	Caracterul nul
\b	Backspace
\B	Orice caracter care nu este la marginea unui cuvânt
\b	Începutul sau sfârșitul unui cuvânt (dacă nu este cuprins între paranteze)
\cX	Expresia Ctrl-X (X fiind orice caracter din alfabet)
\d	O singură cifră
\D	Orice caracter care nu este cifră
\f	Caracterul salt la pagină nouă
\n	Caracterul salt la linie nouă
\ooo	Valoarea octală reprezentată de cifrele ooo (fiecare „o” reprezintă o cifră între 0 și 7)
\r	Caracterul retur de car
\s	Orice caracter care nu este un spațiu
\s	Orice caracter care reprezintă un spațiu (spațiu, tab sau linie nouă)
\t	Caracterul de tabulare
\w	Orice caracter care nu este alfanumeric
\w	Orice caracter alfanumeric
\xhh	Valoarea hexazecimală reprezentată de cifrele hh (fiecare „h” reprezintă o cifră hexazecimală între 0 și f)
^	Începutul unei linii

Pentru a detecta unul dintre caracterele speciale `$`, `|`, `*`, `^`, `[`, `]`, `\` sau `/`, trebuie ca acestea să fie precedate de caracterul backslash (`\`). În consecință, aceste caractere vor apărea în expresiile regulate sub forma `\$`, `\|`, `\*`, `\^`, `\[`, `\]`, `\\` și respectiv `\/`. Expresiile regulate par adesea

complicate din cauza preponderenței unor secvențe ciudate de caractere și utilizării caracterelor backslash. Ca și în alte cazuri, puteți începe prin a utiliza numai câteva dintre aceste facilități.

Până acum, această secțiune a prezentat doar sintaxa expresiilor regulate, fără a descrie utilizarea acestora în Perl. De regulă, expresiile regulate sunt incluse între două caractere slash (/ /), utilizându-se operatorii de echivalență (==) sau de neechivalență (!~) pentru a verifica un anumit șir de caractere. Puteți scrie cu ușurință un script în Perl care să caute aceeași expresie pe care am căutat-o cu ajutorul comenzii grep la începutul acestei secțiuni. Pentru aceasta, procedați după cum urmează:

1. Utilizați un editor de texte pentru a salva următorul script într-un fișier numit lookup:

```
#!/usr/bin/perl

while(<STDIN>)
{
    if($_ =~ /[bB]laster/ ) { print $_; }
}
```

2. Acordați fișierului lookup dreptul de execuție, utilizând comanda:

```
chmod +x lookup
```

3. Încercați să rulați scriptul, utilizând comanda:

```
cat /usr/src/linux/drivers/sound/sbpcd.c | lookup
```

Iată, spre exemplu, ce poate afișa sistemul dumneavoastră:

```
* Works with SoundBlaster compatible cards and with "no-sound"
0x230, 1, /* SoundBlaster Pro and 16 (default) */
0x250, 1, /* OmniCD default, SoundBlaster Pro and 16 */
0x270, 1, /* SoundBlaster 16 */
0x290, 1, /* SoundBlaster 16 */
static const char *str_sb = "SoundBlaster";
static const char *str_sb_1 = "soundblaster";
* sbpcd=0x230,SoundBlaster
msg(DBG_INF, " LIL0 boot:...
Sbpcd=0x230,SoundBlaster\n");
```

Comanda cat furnizează conținutul unui anumit fișier (care, după cum știți din exemplul cu comanda grep, cuprinde anumite linii care conțin expresia regulată) scriptului de căutare. Acesta utilizează operatorul din Perl de echivalență a expresiilor regulate (==) și afișează toate liniile echivalente.

Este necesar să explicăm maniera în care am utilizat variabila \$_ în acest script. Expresia <STDIN> citește o linie de la intrarea standard, pe care o memorează în mod prestabilit în variabila \$_. În ciclul while se testează echivalența dintre expresia regulată și variabila \$_. Întreaga acțiune a scriptului lookup este de fapt executată de o singură instrucțiune:

```
if($_ =~ /[bB]laster/ ) { print $_; }
```

Acest exemplu demonstrează modul în care puteți utiliza o expresie regulată pentru a căuta aparițiile unor șiruri de caractere într-un fișier.

După ce veți utiliza pentru o vreme expresii regulate, veți putea estima mai bine potențialul acestora. Esențialul este să stabiliți corect expresia regulată care corespunde modelului de care aveți nevoie. Iată, spre exemplu, un script care caută toate liniile care încep cu exact șapte spații și se termină cu o paranteză închisă:

```
while(<STDIN>)
{
    if($_ =~ /\)\n/ && $_ =~ /\^ {7}\S/ ) { print $_; }
}
```

Instrucțiuni de control

Până acum, ați văzut instrucțiuni Perl care se execută în manieră secvențială, adică una după cealaltă. Limbajul Perl conține de asemenea instrucțiuni care vă permit să modificați fluxul execuției instrucțiunilor. Deja ați văzut instrucțiunea if și ciclul while. În Perl este implementat un set complet de instrucțiuni de control, întocmai ca și în C, chiar cu unele completări.

Toate instrucțiunile condiționale au în Perl forma:

```
instrucțiune-condițională
{ cod Perl care se execută atunci când condiția este adevărată }
```

Observați că trebuie să includeți în paranteze codul care urmează unei instrucțiuni condiționale. Instrucțiunea condițională va calcula valoarea unei expresii pentru a decide dacă execută sau nu codul din paranteze. În Perl, ca și în C, orice valoare nenulă corespunde valorii „adevărat”, în timp ce o valoare nulă reprezintă valoarea „fals”.

Următoarele secțiuni descriu pe scurt sintaxa principalelor instrucțiuni condiționale din Perl.

if și unless

Instrucțiunea if din Perl este similară cu instrucțiunea if din C. Spre exemplu, iată o instrucțiune if care verifică dacă un anumit contor nu a depășit un prag:

```
if( $count > 25 ) { print "Prea multe erori !\n"; }
```

Instrucțiunea if permite adăugarea unei clauze else, ca în exemplul următor:

```
if( $user eq "root" )
{
    print "Simularea incepe...\n";
}
else
{
    print "Stimate $user, trebuie sa fiti \"root\" pentru a putea rula acest
program.\n";
    exit;
}
```

Cunoscătorii limbajului C vor observa imediat asemănarea în ceea ce privește sintaxa între C și Perl. Condiționalele construite cu instrucțiunea if pot fi urmate de zero sau mai multe clauze elsif, care pot acoperi mai multe alternative, ca în exemplul care urmează:

```
print "Numarul versiunii: "; # cerem utilizatorului
                             # numarul versiunii
$os_version = <STDIN>; # citim de la intrarea standard
chop $os_version;      # eliminam caracterul "newline"
                       # de la sfarsitul liniei

# Verificarea numarului versiunii
if($os_version >= 10) { print "Nu este necesara actualizarea\n"; }
elsif ($os_version >= 6 && $os_version < 9)
{ print "Actualizare standard\n"; }
elsif ($os_version > 3 && $os_version < 6)
{ print "Este necesara o reinstalare\n"; }
else { print "Ne pare rau, nu putem efectua actualizarea\n"; }
```

Instrucțiunea unless este specifică limbajului Perl. Această instrucțiune are aceeași formă ca și if și permite inclusiv clauzele elsif și else. Diferența este că unless își va executa corpul numai atunci când condiția este falsă. Iată de exemplu cum se poate rescrie unul din exemplele anterioare folosind unless:

```
unless ($user eq "root")
{
```

```
print "Trebuie sa fiti \"root\" pentru a putea rula acest program.\n";
exit;
}
```

În acest caz, scriptul va continua numai dacă șirul de caractere `user` are valoarea „root”; altfel, scriptul se va termina.

Ciclul while

Puteți utiliza instrucțiunea `while` pentru *ciclare*, adică pentru repetarea unor procesări până când o anumită condiție devine falsă. Spre exemplu, iată cum putem citi intrarea standard linie cu linie, prelucrând separat fiecare linie citită:

```
while ($in = <STDIN>)
{
    # Cod care prelucreaza linia
    print $in;
}
```



Atunci când citiți de la intrarea standard fără ca să specificați nici un argument, linia curentă din intrarea standard este în mod prestabilit atribuită variabilei `$_`. Prin urmare, ciclul precedent poate fi astfel rescris într-o formă mai simplă:

```
while (<STDIN>)
{
    # Cod care prelucreaza linia
    print $_;
}
```

Instrucțiunea `while` are în Perl o utilitate mult mai mare decât în C, condițiile având în Perl forme mult mai variate. Spre exemplu, dacă utilizați un tablou în locul unei condiții, ciclul `while` se va executa până la epuizarea tuturor elementelor din tablou, după cum se poate observa și din următorul exemplu:

```
# Presupunem ca in tabloul @arg se gasesc optiunile din linia de comanda
while (@arg)
{
    $arg = shift @arg; # extragem un parametru
    # Cod care prelucreaza parametrul curent
    print $arg;
}
```

Funcția `shift` șterge primul element dintr-un tablou, returnând elementul șters.

Putem „sări” direct la sfârșitul unui ciclu utilizând cuvântul cheie `next`; cuvântul cheie `last` determină ieșirea forțată din ciclu. Ciclul `while` prezentat în continuare adună numerele de la 1 la 10, cu excepția lui 5:

```
while (1)
{
    $i++;
    if($i == 5) { next; } # daca $i este 5, sarim la
    # iteratia urmatoare
    if($i > 10) { last; } # cand $i depaseste 10,
    # ciclul se termina
    $sum += $i; # adauga numarul $i curent la suma
}
# In acest moment, $sum trebuie sa aiba valoarea 50
```

for și foreach

Instrucțiunea `for` are în Perl o sintaxă similară cu cea utilizată în C. În general, `for` se utilizează pentru a executa o anumită instrucțiune de un număr oarecare de ori, în funcție de valoarea unei anumite expresii. Sintaxa acestei instrucțiuni este următoarea:

```
for(expr_1; expr_2; expr_3) { corp }
```

Expresia `expr_1` este evaluată o singură dată, la începutul ciclului, iar corpul ciclului se execută până când expresia `expr_2` are valoarea zero. Cea de-a treia expresie, `expr_3`, se evaluează după fiecare execuție a corpului ciclului. Oricare dintre cele trei expresii poate lipsi, dar fără a omite caracterul delimitator punct și virgulă („;”). De asemenea, corpul instrucțiunii trebuie inclus între acolade chiar și în cazul în care este format dintr-o singură instrucțiune. Iată un exemplu de ciclu `for` care realizează adunarea numerelor de la 1 la 10:

```
for($i=0, $sum=0, $i<=10; $sum+=$i, $i++) { }
```

În acest exemplu, operația de adunare a numerelor este efectuată datorită evaluării celei de-a treia expresii, corpul propriu-zis al instrucțiunii `for` fiind vid (`{ }`).

Instrucțiunea `foreach` se utilizează pentru prelucrarea tablourilor. Sintaxa acesteia este următoarea:

```
foreach Variabila (Tablou) { corp }
```

Instrucțiunea `foreach` atribuie variabilei *Variabila* un element din tabloul *Tablou* și execută corpul. Această operație se repetă până când în tablou nu mai rămâne nici un element. Iată cum putem aduna numerele de la 1 la 10 utilizând o instrucțiune `foreach`:

```
foreach $i (1..10) { $sum += $i; }
```

Observați că tabloul a fost definit prin intermediul operatorului de definire a unui domeniu (`..`). Tabloul poate fi de asemenea reprezentat printr-o listă de elemente, separate prin virgule.



Dacă omiteți să precizați variabila într-o instrucțiune `foreach`, interpretorul perl va memora pe rând fiecare element al tabloului în variabila predefinită `$_`. Prin urmare, instrucțiunea de mai sus se poate scrie în următoarea formă simplificată:

```
foreach (1..10) { $sum += $_; }
```

goto

Instrucțiunea `goto` transferă necondiționat controlul la o etichetă de instrucțiune. Iată în continuare un exemplu care cere utilizatorului să introducă o valoare, repetând cererea până când valoarea introdusă este acceptabilă:

```
ReEnter:
print "Introduceți deplasamentul: ";
$offset = <STDIN>;
chop $offset;
unless ($offset > 0 && $offset < 512)
{
    print "Deplasament necorespunzător: $offset\n";
    goto ReEnter;
}
```

Accesul la comenzile sistemului Linux

Puteți executa orice comandă a sistemului Linux dintr-un program Perl în mai multe moduri:

- Apelând funcția `system` cu un șir care reprezintă chiar comanda pe care doriți să o executați.

- Incluzând o comandă Linux între *apostrofuri inverse* (```), cunoscute și sub numele de *accente grave*. În acest mod, puteți rula o comandă Linux, obținând totodată și rezultatele pe care aceasta le afișează.
- Apelând funcția `fork` pentru a copia scriptul curent și pentru a executa ulterior noi comenzi în procesul copil creat (atunci când un proces lansează un alt proces, acesta este cunoscut sub numele de *proces copil* al primului).
- Apelând funcția `exec` pentru a suprascrie scriptul curent cu un alt script sau cu o comandă Linux.
- Utilizând combinat funcțiile `fork` și `exec` pentru a simula comportamentul interpretorului (monitorizând datele introduse de utilizator de la tastatură și prelucrând fiecare comandă introdusă cu ajutorul unui proces copil). În secțiunea de față, vom prezenta un exemplu simplu prin care vom arăta cum se poate realiza această operație.

Cel mai simplu mod de a executa o comandă Linux într-un script Perl este utilizând funcția `system`, comanda respectivă fiind inclusă într-un șir. După terminarea execuției funcției `system`, codul de retur al comenzii executate se găsește în variabila `$?` . Puteți scrie cu ușurință un script Perl care citește un șir de caractere de la intrarea standard și procesează acest șir cu ajutorul funcției `system`. Pentru aceasta, procedați după cum urmează:

1. Utilizați un editor de texte pentru a scrie și apoi salva următorul script într-un fișier numit `rcmd.pl`:

```
#!/usr/bin/perl
# Citeste datele introduse de utilizator si executa comanda

$prompt = "Introduceti comanda (\`exit\` pentru terminare): ";
print $prompt;

while (<STDIN>)
{
    chop;
    if($_ eq "exit") { exit 0; }
    # Executa comanda apeland functia system
    system $_;
    unless ($? == 0) { print "Eroare la executia comenzii: $_\n"; }
    print $prompt;
}
```

2. Acordați fișierului `rcmd.pl` drept de execuție, utilizând comanda:

```
chmod +x rcmd.pl
```

3. Rulați scriptul, tastând `rcmd.pl` la promptul interpretorului. Prezentăm mai jos un exemplu de rezultat pe care scriptul îl poate afișa (acesta depinde de comanda introdusă):

```
Introduceti comanda (\`exit\` pentru terminare): ps
PID TTY STAT  TIME  COMMAND
 415 p0  S    0:00  /bin/login -h lnb486 -p
 416 p0  S    0:00  -bash
 585 p0  S    0:00  perl ./rcmd.pl
 586 p0  R    0:00  ps
Introduceti comanda (\`exit\` pentru terminare): exit
```

O altă posibilitate de a rula comenzi UNIX din Perl este utilizând funcțiile `fork` și `exec` în scriptul dumneavoastră. Prezentăm în continuare scriptul numit `psh.pl`, care utilizează `fork` și `exec` pentru a executa comenzile introduse de utilizator:

```
#!/usr/bin/perl
```

```
# Acesta este un script simplu care utilizeaza "fork"
```

```
# si "exec" pentru a rula comenzile introduse
# de utilizator

$prompt = "Introduceti comanda (\`exit\`
           pentru terminare): ";
print $prompt;

while(<STDIN>)
{
    chop;      # elimina caracterul "newline"
    if($_ eq "exit") { exit 0; }

    $status = fork;
    if($status)
    {
        # Codul procesului parinte: asteapta
        # terminarea procesului fiu

        wait;
        print $prompt;
        next;
    }
    else
    {
        exec $_;
    }
}
```

Exemplul următor prezintă modul în care script-ul `psh.pl` execută comanda `ps`:

```
Introduceti comanda (\`exit\` pentru terminare): ps
PID TTY STAT  TIME  COMMAND
 415 p0  S    0:00  /bin/login -h lnb486 -p
 416 p0  S    0:00  -bash
 589 p0  S    0:00  perl ./psh.pl
 590 p0  R    0:00  ps
Introduceti comanda (\`exit\` pentru terminare): exit
```

Interpretoarele UNIX, printre care și Bash, utilizează mecanismul `fork-exec` pentru a rula fiecare comandă.

Accesul la fișiere

Ați remarcat desigur expresia `<STDIN>` în cadrul unor exemple din acest capitol. Aceasta este maniera în care se efectuează citirea datelor dintr-un fișier în Perl. Fișierele sunt identificate în Perl printr-un *măner* (*handle*) de fișier, care este un alt nume pentru identificatorul fișierului. De regulă, mânerele de fișier sunt scrise cu litere mari. `STDIN` este un măner predefinit, asociat intrării standard – în mod prestabilit, aceasta este tastatura sistemului. Alte două mâner predefinite sunt `STDOUT` și `STDERR`. Mănerul `STDOUT` permite afișarea datelor pe terminal, iar `STDERR` este utilizat pentru afișarea mesajelor de eroare.

Pentru a citi date dintr-un fișier, se include mănerul fișierului între paranteze unghiulare (`< >`). Prin urmare, expresia `<STDIN>` realizează citirea unei linii de la intrarea standard.

Puteți deschide și alte fișiere, utilizând funcția `open`. Următorul exemplu deschide fișierul `/etc/passwd` și afișează liniile din acest fișier:

```
open (PWFH, "/etc/passwd"); # indicatorul de fisier este PWFH
while (<PWFH>) { print $_; } # implicit, linia citita este memorata in $_
close PWFH;                # închiderea fișierului
```

În mod prestabilit, funcția `open` deschide un fișier pentru citire. Puteți însă adăuga anumite caractere speciale la începutul numelui fișierului, pentru a indica alte tipuri de acces. Prefixul `>`

permite deschiderea unui fișier pentru scriere, în timp ce prefixul >> permite deschiderea pentru inserția datelor la sfârșitul fișierului. În continuare, iată un script scurt care citește fișierul /etc/passwd și creează un nou fișier, numit output, cu lista tuturor utilizatorilor fără interpretor (liniile corespunzătoare acestor utilizatori în fișierul de parole se termină cu caracterul :) :

```
#!/usr/bin/perl
# Citește fișierul /etc/passwd și generează
# lista utilizatorilor fără shell

open (PWDFILE, "/etc/passwd");
open (RESULT, ">output");

while (<PWDFILE>)
{
    if($_ =~ /\:$/n/) {print RESULT $_: }
}

close PWDFILE;
close RESULT;
```

După execuția acestui script, în directorul curent va apărea un fișier numit output. Iată ce conține de regulă acest fișier după rularea scriptului pe un sistem Linux:

```
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
Nobody:*:99:99:Nobody:/:
```



Un prefix interesant este *caracterul de redirectare (pipe)* – bara verticală (|). Apelând open cu un nume de fișier care începe cu acest caracter, restul numelui fișierului va fi interpretat ca o comandă. Interpretorul Perl execută comanda; puteți utiliza apeluri la funcția print pentru a furniza date de intrare acestei comenzi. Spre exemplu, următorul script Perl trimite un anumit mesaj prin poșta electronică unei liste de utilizatori:

```
#!/usr/bin/perl
# Trimite un mesaj unei liste de utilizatori

foreach ("root", "naba")
{
    open(MAILPIPE, "| mail -s Greetings $_");
    print MAILPIPE "Aveți de terminat raportul săptămânal
de activitate !\n";
    close MAILPIPE;
}
```

Dacă numele unui fișier se termină cu caracterul de redirectare (|), acel fișier va fi executat ca o comandă. Rezultatele afișate de acea comandă pot fi citite utilizând paranteze unghiulare, după cum se observă în exemplul următor:

```
open (PPIPE, "ps -ax |");
while (<PPIPE>)
{
    # Prelucraza fiecare linie afisata de comanda ps -
    # in acest exemplu, afiseaza linia
    print $_;
}
```

Subrutine

Chiar dacă Perl include un număr mare de funcții predefinite, puteți adăuga propriile dumneavoastră module de cod, sub formă de subrutine. De fapt, însăși distribuția limbajului Perl conține un număr mare de subrutine. Iată un exemplu simplu, care prezintă sintaxa subrutinelor în Perl:

```
#!/usr/bin/perl
sub hello
{
    # Efectueaza copii locale ale parametrilor din
    # tabloul @_

    local ($first, $last) = @_;

    print "Hello, $first $last\n";
}

$a = Ion;
$b = Popescu;
```

&hello(\$a, \$b); # Apeleaza subrutina

La rularea scriptului, acesta va afișa următorul rezultat:
Hello, Ion Popescu

Iată câteva amănunte importante legate de subrutine:

- Orice subrutină își primește parametrii în tabloul @_ (simbolul „at”, urmat de liniuța de subliniere).
- Variabilele utilizate într-o subrutină sunt în mod prestabilit globale. Utilizați funcția local pentru a crea variabile locale.
- Apelul unei subrutine se efectuează prin plasarea unui caracter ampersand (&) înainte de numele subrutinei. În consecință, subrutina hello va fi apelată ca &hello.

Dacă doriți, puteți salva fiecare subrutină într-un fișier separat. Subrutina hello, de exemplu, poate fi conținută într-un fișier separat numit hello.pl. Atunci când salvați o subrutină într-un fișier separat, amintuiți-vă să adăugați un cod de retur pe ultima linie din fișier – de exemplu, puteți tasta **1** pe ultima linie pentru a întoarce valoarea 1. Fișierul hello.pl va avea deci următorul conținut:

```
sub hello
{
    # Efectueaza copii locale ale parametrilor din
    # tabloul @_

    local ($first, $last) = @_;

    print "Hello, $first $last\n";
}
1; # codul de retur
```

Iată în continuare un exemplu de program care utilizează subrutina hello:

```
#!/usr/bin/perl
require "hello.pl"; # include fișierul cu
# definiția subrutinei

$a = Ion;
$b = Popescu;

&hello($a, $b); # Apeleaza subrutina
```

Scriptul utilizează funcția `require` pentru a include fișierul `hello.pl` care conține definiția subrutinei `hello`.

Funcții înglobate în Perl

În Perl există aproape 200 de funcții înglobate (denumite și *funcții Perl*), atât unele similare celor din biblioteca runtime C, cât și altele cu acces direct la sistemul de operare. Se recomandă să parcurgeți lista acestor funcții pentru a vă forma o idee despre varietatea și puterea facilităților de care dispune limbajul Perl. Capitolul de față nu dispune de suficient spațiu pentru a prezenta și aceste funcții. Mai multe despre aceste funcții se pot afla de la adresa Web: <ftp://ftp.digital.com/pub/plan/perl/CPAN/doc/manual/html/perlfunc.html>

Rezumat

În esență, sistemul Linux face parte din familia sistemelor UNIX. Pentru a executa multe activități obișnuite în sistem, este necesar să învățați să folosiți un shell (interpretor de comenzi). Chiar dacă dispuneți de o interfață grafică, tot trebuie să deschideți o fereastră `xterm` în care să introduceți comenzi de la prompt-ul interpretorului. Capitolul de față prezintă shellul Bash – Bourne Again Shell – și limbajul Perl.

După lectura acestui capitol, ați învățat următoarele:

- ▶ Chiar și atunci când lucrați tot timpul într-un mediu grafic cum este X Window System, trebuie să tastați comenzi Linux într-o fereastră `xterm` pentru a executa majoritatea activităților obișnuite.
- ▶ Un interpretor este un program care vă permite să rulați comenzi. Interpretorul prestabilit în Linux este Bash. Bash este compatibil cu shellul Bourne, care este utilizat în mod prestabilit în alte sisteme UNIX.
- ▶ Structura de directoare din Linux este organizată logic într-un arbore unic, indiferent de locația fizică a subdirectoarelor. Sistemul Linux oferă multe comenzi pentru lucrul cu fișiere și directoare.
- ▶ Comanda `find` reprezintă o modalitate puternică de a căuta toate fișierele care corespund anumitor criterii.
- ▶ Un script de interpretor este o secvență de comenzi Linux, incluse într-un același fișier. De regulă, este necesar să introduceți o linie specială la începutul fișierului care reprezintă scriptul și apoi să acordați fișierului dreptul de execuție. Scriptul poate fi apoi rulat tastându-i numele la promptul interpretorului.
- ▶ Perl este un limbaj de comandă foarte răspândit, care poate fi instalat de pe CD-ROM-ul anexat acestei cărți. Utilizând Perl, puteți scrie scripturi puternice pe sistemul dumneavoastră.
- ▶ Facilitățile limbajului Perl sunt în mare parte comune cu ale altor limbaje de programare, cum ar fi limbajul C. Printre facilitățile puternice specifice limbajului Perl se numără posibilitatea utilizării expresiilor regulate și căutării unui anumit șablon.

Capitolul 7

Secretele sistemului DOS sub Linux

În acest capitol:

- ▶ Veți accesa o partiție DOS din Linux
- ▶ Veți utiliza programele `mtools` pentru accesul la dischete DOS
- ▶ Veți afla câteva dintre facilitățile emulatorului DOS DOSEMU

De obicei, sistemul Linux se instalează pe un PC pe care anterior au fost configurate sistemele DOS și Microsoft Windows. Dacă trebuie să lucrați atât sub DOS sau Windows, cât și în Linux, aveți nevoie de o modalitate de acces la fișierele DOS din Linux. Acest capitol descrie modul în care puteți monta și utiliza discuri MS-DOS, inclusiv dischete.

Veți mai învăța să lucrați cu un pachet de programe numit `mtools`, care vă permite să aveți acces și să utilizați (operațiile permise fiind copierea, ștergerea și formatarea) fișiere MS-DOS (aflate în majoritatea cazurilor pe o dischetă) din Linux.

Montarea unui sistem de fișiere

Dacă MS-DOS și Microsoft Windows sunt instalate pe o altă partiție a hard-discului, este foarte posibil ca acea partiție să fie montată și sub Linux. La instalare (vezi capitolul 1), programul de instalare al sistemului Red Hat va rula utilitarul Disk Druid, care vă întreabă dacă doriți să aveți acces de sub Linux la partițiile DOS existente. Programul Disk Druid detectează aceste partiții (ca și partițiile OS/2) verificând tabela de partiții a hard-discului.

Utilizând programul Disk Druid, puteți specifica locul în care doriți să montați fiecare partiție DOS. (În urma operației de montare, ierarhia de directoare DOS va apărea ca făcând parte din sistemul de fișiere Linux.) În majoritatea cazurilor, prima partiție DOS este montată ca `/dos`, cea de-a doua ca `/dosd` etc. Specificând aceste locuri de montare, Disk Druid efectuează toți pașii necesari pentru a se asigura că partițiile DOS sunt automat montate la fiecare pornire a sistemului Linux.



Pentru a verifica dacă partiția DOS a discului dumneavoastră a fost automat montată, parcurgeți următoarele etape (nu este necesar să lucrați ca root):

1. Utilizați comanda `grep` pentru a căuta șirul de caractere `msdos` în fișierul `/etc/fstab`. Iată ce vă poate afișa sistemul:

```
grep msdos /etc/fstab
/dev/hda1 /dosd msdos defaults 0 0
```

Vom prezenta semantica liniilor din fișierul `/etc/fstab` în secțiunea „Fișierul `/etc/fstab`” din acest capitol.

2. Dacă apare cel puțin o linie care conține șirul de caractere `msdos`, înseamnă că sistemul Linux deja montează automat partiții DOS. În exemplul de mai sus, se observă că linia care conține modelul `msdos` are pe primul câmp numele partiției `/dev/hda1` (prima partiție a discului IDE); cel de-al doilea câmp, `/dos`, reprezintă locul în care va fi montată partiția.

3. Dacă însă comanda `grep` nu arată nici o linie care să conțină șirul de caractere `msdos` în fișierul `/etc/fstab`, înseamnă că sistemul dumneavoastră nu montează automat nici o partiție DOS a discului. O explicație ar fi că hard-discul dumneavoastră nu conține partiții DOS.



Sugestie

O altă modalitate rapidă de a detecta dispozitivele montate este să tastați comanda `mount` (fără parametri) la promptul interpretorului. Un exemplu de rezultat afișat de `mount` este următorul:

```
/dev/hda3 on / type ext2 (rw)
/dev/hda1 on /dos type msdos (rw) (Partitia MS-DOS montata in /dos)
```

Linii care conțin șirul de caractere `msdos` indică sistemele de fișiere MS-DOS montate sub Linux. În cazul de mai sus, se montează automat o singură partiție DOS în directorul `/dos`.

Secțiunile următoare explică mecanismul de montare automată a partițiilor DOS. Chiar dacă discul dumneavoastră nu are partiții DOS, este bine să știți cum puteți avea acces la un sistem de fișiere DOS de sub Linux, pentru că la un moment dat va fi necesar să lucrați cu o dischetă DOS sub Linux. Pentru a cunoaște lucrul cu sistemele de fișiere DOS sub Linux, este foarte important să înțelegeți conceptul de montare.

Comanda mount

După cum s-a precizat și în capitolul 6, Linux are un singur sistem de fișiere, care începe direct din directorul rădăcină, notat cu `/`. Chiar dacă aveți un al doilea disc (sau mai multe partiții pe un singur disc), conținutul acestuia apare amplasat din punct de vedere logic în altă parte în sistemul de fișiere Linux. *Montarea* este operația în urma căreia un anumit dispozitiv fizic de memorare a datelor (care poate fi o partiție a unui disc fix sau un CD-ROM) va apărea ca făcând parte din sistemul de fișiere Linux.

Multe sisteme Linux au doar o partiție mică a discului montată în directorul rădăcină (`/`) și o partiție ceva mai mare în directorul `/usr`. Acest director necesită o partiție ceva mai mare, deoarece multe pachete software, inclusiv X Window System, vor fi instalate în directorul `/usr`.

Comanda `mount` poate fi utilizată și pentru a monta manual un anumit dispozitiv în sistemul de fișiere Linux într-un director precizat. Acest director este denumit *punct de montare* (*mount point*). Orice director poate fi utilizat ca punct de montare. Dacă montați însă un dispozitiv într-un director care nu este vid, nu veți mai avea acces la fișierele din acel director până când nu veți demonta dispozitivul utilizând comanda `umount`. Prin urmare, se recomandă ca întotdeauna să folosiți un director vid (cum ar fi `/mnt`) ca punct de montare.

Ca multe alte comenzi UNIX, și `mount` are foarte multe opțiuni. Pentru început, vă puteți descurca reținând numai câteva dintre acestea.



Observație

Din cauză că montarea face ca un dispozitiv fizic să devină parte a sistemului Linux, în mod prestabil numai utilizatorul `root` este autorizat să execute comanda `mount`. Utilizatorul `root` poate însă configura anumite linii din fișierul `/etc/fstab` pentru a permite fiecărui utilizator să monteze anumite dispozitive. Consultați secțiunea „Montarea dischetelor DOS” pentru a vedea modul de a atribui oricărui utilizator permisiunea

de a monta dischete DOS. Dacă încercați să montați un anumit dispozitiv atunci când nu ați intrat în sistem ca `root` și primiți mesajul:

```
mount: only root can do that
```

înseamnă că trebuie mai întâi să intrați în sistem ca `root`.



Dacă nu ați intrat ca `root`, puteți utiliza comanda `su` pentru a deveni `root` fără a deschide o nouă sesiune. Atunci când tastați `su` fără nici un parametru, interpretorul presupune că doriți să deveniți `root` și vă cere parola de `root`, după cum urmează:

```
[nab@alntp200 naba]$ su      (Incercati sa deveniti root)
Password:                    (Introduceti parola de root)
[nab@alntp200 naba]# su      (Ati devenit deja root)
```

După ce ați introdus parola de `root`, promptul se modifică, indicându-vă că ați devenit `root`.

Ca `root`, să presupunem că doriți să montați unitatea de CD-ROM (numele acesteia este `/dev/cdrom`) în punctul de montare `/mnt/cdrom` (care trebuie să existe în sistemul dumneavoastră). Pentru aceasta, tastați comanda:

```
mount /dev/cdrom /mnt/cdrom
```

Comanda `mount` va semnala o eroare dacă unitatea CD-ROM este deja montată. În caz contrar, operația de montare va reuși și veți putea avea acces la conținutul CD-ROM-ului prin intermediul directorului `/mnt/cdrom`.

Pentru a monta o partiție DOS, se utilizează un format similar al comenzii `mount`, dar trebuie precizat în plus tipul sistemului de fișiere de pe partiția DOS. Dacă partiția DOS este pe prima unitate IDE (Integrated Drive Electronics) și doriți să o montați în directorul `/dos`, utilizați următoarea comandă `mount`:

```
mount -t msdos /dev/hda1 /dos
```

Partea `-t msdos` a comenzii `mount` precizează faptul că dispozitivul montat - `/dev/hda1` - conține un sistem de fișiere MS-DOS. Linux conține suport încorporat pentru fișierele MS-DOS. Figura 7-1 prezintă efectul comenzii `mount`.

Tot în figura 7-1 mai puteți observa modul în care directoarele de pe partiția DOS sunt denumite în sistemul de fișiere Linux. Spre exemplu, directorul `C:\DOS` din DOS va deveni `/dos/dos` în Linux. Similar, directorul `C:\WINDOWS` se va numi în Linux `/dos/windows`. Probabil că deja ați dedus modelul general. Pentru a converti numele unui fișier DOS în Linux (în cazul particular în care partiția DOS este montată în directorul `/dos`), trebuie să aplicați următoarele reguli:

- Înlocuiți toate literele mari din numele DOS cu litere mici.
- Înlocuiți `C:\` cu `/dos/`.
- Înlocuiți toate caracterele „backslash” (`\`) cu „slash” (`/`).

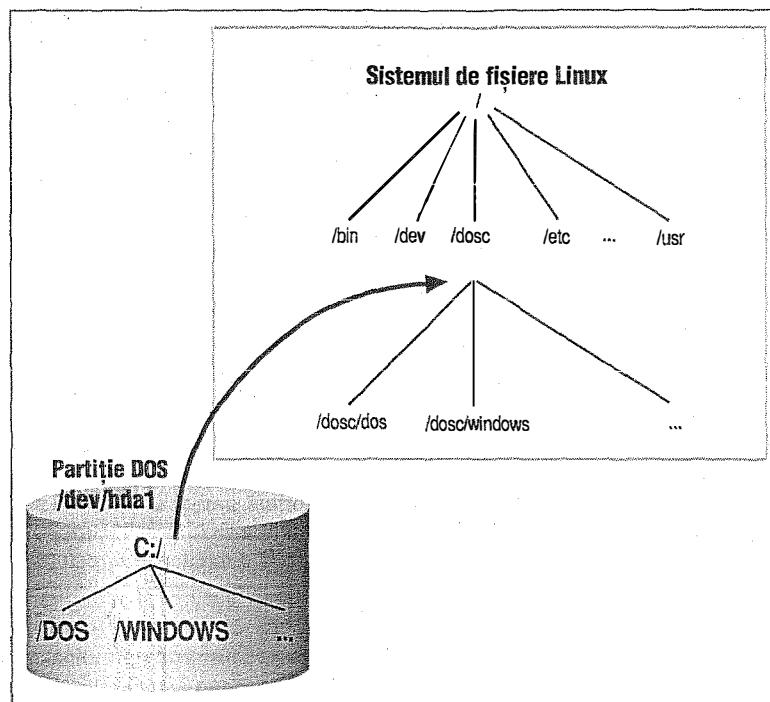
Lucrul cu dischete DOS

Montarea unei dischete DOS într-un sistem Linux este o operație similară cu montarea unei partiții DOS. De regulă, trebuie să intrați în sistem ca `root` pentru a monta o dischetă, dar, parcurgând etapele unui proces pe care îl vom prezenta în finalul acestei secțiuni, vă puteți configura sistemul pentru a permite oricărui utilizator să monteze o dischetă DOS. Trebuie de asemenea să cunoașteți numele dispozitivului corespunzător unității de disc. În mod prestabilit, în Linux se utilizează două nume generice pentru a defini unități de dischetă:

- /dev/fd0, care corespunde unității A (prima unitate de dischetă)
- /dev/fd1, corespunzător unității B (a doua unitate de dischetă, în caz că aceasta există în configurația dumneavoastră)

Figura 7-1

Montarea unei
partiții DOS în
directorul /dos



În ceea ce privește punctul de montare, directorul numit /mnt/floppy este special destinat pentru montarea temporară a dischetelor DOS. Comanda care vă permite montarea unei dischete DOS în directorul /mnt/floppy este următoarea:

```
mount -t msdos /dev/fd0 /mnt/floppy
```

După montarea dischetei, puteți copia fișiere pe dischetă sau de pe dischetă utilizând comanda de copiere din Linux (cp). Pentru a copia de exemplu fișierul xtmenu1.pcx din directorul curent pe dischetă, tastați comanda:

```
cp xtmenu1.pcx /mnt/floppy
```

În mod similar, pentru a vizualiza conținutul dischetei, introduceți următoarea comandă:

```
ls /mnt/floppy
```

Dacă doriți să scoateți discheta din unitate, trebuie mai întâi să demontați unitatea. Această operație elimină asocierea dintre sistemul de fișiere de pe dischetă și punctul de montare din sistemul Linux. Utilizați comanda umount pentru a demonta un dispozitiv, în cazul de față cu sintaxa:

```
umount /dev/fd0
```

Puteți de asemenea să configurați sistemul Linux astfel încât orice utilizator să poată monta o dischetă DOS. Este suficient să intrați în sistem ca root și să adăugați o linie în fișierul

/etc/fstab. De exemplu, pentru a permite utilizatorilor să monteze discheta DOS din unitatea A în directorul /a, parcurgeți următoarele etape:

1. Intrați în sistem ca root.

2. Creați directorul /a, folosind comanda de mai jos:

```
mkdir /a
```

3. Modificați fișierul /etc/fstab cu ajutorul unui editor de texte (cum ar fi vi sau Emacs), inserând linia de mai jos, după care salvați fișierul și părăsiți editorul:

```
/dev/fd0 /a msdos noauto,user0 0
```

Puteți afla mai multe despre fișierul /etc/fstab din secțiunea următoare; opțiunea user (care apare lângă noauto) permite tuturor utilizatorilor să monteze dischete DOS. Primul câmp al liniei corespunde numelui dispozitivului (/dev/fd0), al doilea punctului de montare (/a), iar al treilea tipului sistemului de fișiere (msdos).

4. Ieșiți din sesiunea curentă și intrați în sistem ca utilizator fără privilegii (adică nu root).

5. Pentru a testa dacă puteți monta o dischetă DOS fără a lucra ca root, introduceți o dischetă DOS în unitatea A și tastați următoarea comandă:

```
mount /a
```

Remarcați că singurul parametru al comenzii mount este în acest caz punctul de montare. Operația de montare trebuie să reușească, iar tastând comanda ls /a trebuie să vedeți lista fișierelor de pe discheta DOS.

6. Pentru a demonta discheta DOS, tastați umount /a.

Fișierul /etc/fstab

În Linux, directorul /etc conține o colecție de fișiere text în care se găsesc informații de configurare pentru sistem. După cum ați văzut în capitolul 5, fișierul /etc/inittab conține informațiile referitoare la procesele care sunt lansate imediat după pornirea sistemului. Un astfel de fișier de configurare este și /etc/fstab, care conține informațiile necesare comenzilor mount și umount. Fiecare linie din fișierul /etc/fstab conține informațiile corespunzătoare unui dispozitiv care va fi montat într-un director din sistemul de fișiere Linux.

Iată un exemplu de fișier /etc/fstab într-un sistem Linux obișnuit:

/dev/hda3	/	ext2	defaults	0 0
/dev/hda1	/dos	msdos	defaults	0 0
/dev/hda5	/dosd	msdos	defaults	0 0
/dev/hda4	swap	swap	defaults	0 0
/dev/fd0	/mnt/floppy	ext2	noauto	0 0
/dev/cdrom	/mnt/cdrom	iso9660	noauto,ro	0 0
none	/proc	proc	defaults	0 0

Primul câmp al fiecărei linii conține numele unui dispozitiv, cum ar fi o partiție a hard-discului. Cel de-al doilea câmp reprezintă punctul de montare, iar al treilea indică tipul sistemului de fișiere montat pe acel dispozitiv. Deocamdată putem ignora ultimele trei câmpuri.

Fișierul /etc/fstab de mai sus ne arată că dispozitivul /dev/hda4 (a patra partiție de pe primul disc IDE) este utilizat ca dispozitiv de schimb pentru memoria virtuală; prin urmare, atât punctul de montare, cât și tipul sistemului de fișiere au valoarea swap. Pe ultima linie se găsește un alt sistem special de fișiere – sistemul de fișiere proc – utilizat de sistemul Linux pentru a memora anumite informații. Șirul de caractere msdos apare în coloana corespunzătoare tipurilor sistemelor de fișiere pe linia care precizează că partiția DOS /dev/hda1 trebuie montată în directorul /dos.



Secret

Conținutul fișierului `/etc/fstab` este utilizat pentru montarea automată a diferitelor sisteme de fișiere în Linux. La pornirea sistemului Linux, procesul `init` execută un script de interpretor care apelează comanda `mount` cu opțiunea `-a`. Această opțiune determină comanda `mount` să citească fișierul `/etc/fstab` și să monteze toate sistemele de fișiere listate în fișier (cu excepția celor cu opțiunea `noauto`). Pentru a monta automat o partiție DOS, trebuie deci să adăugați o linie în fișierul `/etc/fstab` care să conțină informațiile necesare pentru montarea partiției. Dacă doriți spre exemplu să montați sistemul de fișiere DOS de pe prima partiție a discului Small Computer System Interface (SCSI) în directorul `/dosd`, adăugați linia următoare în fișierul `/etc/fstab`:

```
/dev/sda1 /dosd msdos defaults0 0
```

Al patrulea câmp al fiecărei linii din fișierul `/etc/fstab` conține o listă de opțiuni separate prin virgule, specifice fiecărui tip de dispozitiv. De regulă, acest câmp conține opțiunea `defaults`. Această opțiune implică, printre altele, faptul că dispozitivul este montat la pornirea sistemului, că numai utilizatorul `root` poate monta dispozitivul, iar acesta este montat pentru a permite atât citirea cât și scrierea. Dacă printre opțiuni se găsește și `noauto`, atunci dispozitivul nu va fi automat montat la pornirea sistemului. O altă opțiune utilă este `user`. Dispozitivele listate în fișierul `/etc/fstab` cu această opțiune pot fi montate de către orice utilizator. De exemplu, dacă doriți să permiteți oricărui utilizator să monteze CD-ROM-ul, intrați ca `root` și adăugați opțiunea `user` la linia `/dev/cdrom` din fișierul `/etc/fstab`, ca în exemplul de mai jos:

```
/dev/cdrom /mnt/cdrom iso9660 noauto,ro,user 0 0
```

Această linie din `/etc/fstab` va permite oricărui utilizator să monteze un CD-ROM utilizând comanda:

```
mount /mnt/cdrom
```

Utilizarea pachetului mtools

Secțiunile anterioare au prezentat o modalitate de acces la un sistem de fișiere MS-DOS: montarea discului sau dischetei DOS utilizând comanda `mount`, urmată de utilizarea unor comenzi Linux obișnuite, cum sunt `ls` și `cp`. Această metodă de montare a unui sistem de fișiere DOS este recomandată în cazul discurilor fixe. Sistemul Linux va monta automat partiția DOS la pornire, astfel încât să puteți avea acces la directoarele DOS oricând aveți nevoie.

Dacă însă doriți să listați rapid fișierele de pe o dischetă DOS, montarea dischetei este o operație greoaie. Mai întâi, trebuie să montați discheta, după care să listați conținutul acesteia cu comanda `ls` și în final să utilizați comanda `umount` înainte de a scoate discheta din unitate.

O modalitate mai eficientă de lucru cu dischetele DOS este oferită de pachetul de programe `mtools`. Pachetul `mtools` implementează câteva dintre cele mai frecvent utilizate comenzi DOS. Aceste comenzi au aceleași nume ca și în DOS, fiecare comandă fiind însă prefixată cu litera `m` (de la MS-DOS). Prin urmare, comanda cu care puteți lista conținutul unui director este `mdir`, în timp ce `mcopy` realizează copierea unor fișiere. Cel mai important avantaj al pachetului `mtools` este că nu mai este nevoie să montați discheta pentru a utiliza comenzile `mtools`.



Trăitor
Din cauză că programele `mtools` citesc și scriu direct de pe dispozitivul fizic (dischetă), trebuie să intrați în sistem ca `root` pentru a utiliza aceste comenzi. Dacă doriți ca orice utilizator să poată folosi aceste comenzi, trebuie să modificați permisiunile de acces ale dispozitivelor corespunzătoare unităților de disc. Utilizați comanda următoare pentru a permite oricărui utilizator să citească și să scrie pe prima unitate de dischetă:

```
chmod o+rw /dev/fd0
```

Este pachetul mtools instalat?

Pachetul `mtools` este inclus în distribuția sistemului Red Hat Linux de pe CD-ROM-ul anexat acestei cărți. La instalarea sistemului Linux, pachetul `mtools` a fost automat instalat ca parte de bază a sistemului de operare. Fișierele executabile din pachetul `mtools` se găsesc în directorul `/usr/bin`. Pentru a verifica dacă pachetul `mtools` este instalat, tastați `ls /usr/bin/mdir` la promptul de comandă al shellului. Dacă fișierul este detectat de comanda `ls`, înseamnă că pachetul `mtools` este disponibil în sistemul dumneavoastră.

Puteți să verificați dacă `mtools` este instalat în sistem și cu ajutorul următoarei comenzi `rpm`:

```
rpm -q mtools
mtools-3.6-4
```

Dacă `mtools` este instalat, această comandă va afișa numele complet al pachetului `mtools`. În exemplul de mai sus, versiunea instalată în sistem este 3.6.

Pentru a încerca efectiv `mtools`, procedați în modul următor:

1. Intrați în sistem ca `root`, sau tastați `su`, după care introduceți parola de `root`.
2. Introduceți o dischetă MS-DOS în unitatea A a sistemului dumneavoastră.
3. Tastați `mdir`. Comanda va lista conținutul directorului de pe dischetă, în format MS-DOS standard.

Fișierul /etc/mtools.conf

Pachetul `mtools` funcționează de regulă cu configurația inițială, dar în cazul apariției anumitor erori trebuie să verificați fișierul `/etc/mtools.conf`. Acest fișier conține definițiile unităților (cum ar fi A, B sau C) cu care aceste utilitare pot lucra. Iată câteva linii dintr-un fișier `/etc/mtools.conf` obișnuit:

```
drive a: file="/dev/fd0" exclusive
drive b: file="/dev/fd1" exclusive
```

```
# First IDE hard disk partition
drive c: file="/dev/hda1"
```

```
# First SCSI hard disk partition
# drive c: file="/dev/sda1"
```

Semnul diez (#) indică liniile de comentariu. Fiecare linie din fișier definește o anumită unitate, numele dispozitivului asociat acelei unități și un set de cuvinte cheie care indică modul în care se efectuează accesul la dispozitiv. În acest exemplu, primele două linii definesc unitățile A și B. A treia linie care nu este comentariu definește unitatea C ca fiind prima partiție de pe prima unitate IDE (`/dev/hda1`). Dacă aveți și alte unități DOS (de exemplu, D), puteți adăuga și alte linii care definesc în mod corespunzător aceste unități.

În cazul în care unitatea A a sistemului dumneavoastră este o unitate de înaltă densitate de 3,5 inch, nu trebuie să modificați nimic în fișierul inițial `/etc/mtools.conf`.

De regulă, utilitarele `mtools` se folosesc pentru accesarea dischetelor. Chiar dacă puteți defini unitățile C și D corespunzătoare partițiilor DOS de pe discul dumneavoastră, se recomandă să lucrați cu aceste partiții montându-le mai întâi cu comanda `mount`. Datorită montării automate a partițiilor discului fix la pornire, accesul la aceste partiții prin intermediul comenzilor Linux se efectuează foarte ușor.

Comenzile mtools

După cum am precizat anterior în acest capitol, pachetul `mtools` este o colecție de utilitare. Până acum, am văzut cum lucrează comanda `mdir` – corespondentul în `mtools` al comenzii `DIR` din DOS.



Sugestie

Dacă știți să lucrați cu comenzile MS-DOS, vă veți descurca ușor cu comenzile `mtools`. Este suficient să tastați comanda DOS cu litere mici și să vă amintiți să adăugați litera `m` în fața fiecărei comenzi. Deoarece numele comenzilor și fișierelor din Linux sunt sensibile la diferența dintre majuscule și literele mici, trebuie să tastați numele comenzilor `mtools` utilizând numai litere mici.

În tabelul 7-1 puteți găsi toate cele 13 comenzi `mtools`.

Tabelul 7-1 Comenzile mtools

Utilitar mtools	Comandă DOS	Efect
<code>mattrib</code>	ATTRIB	Modifică atributele fișierelor MS-DOS
<code>mcd</code>	CD	Schimbă directorul curent MS-DOS
<code>mcopy</code>	COPY	Copiază fișiere între MS-DOS și Linux
<code>mdel</code>	DEL sau ERASE	Șterge un fișier MS-DOS
<code>mdir</code>	DIR	Listează conținutul unui director MS-DOS
<code>mformat</code>	FORMAT	Copiază un sistem de fișiere MS-DOS pe o dischetă formatată la nivel scăzut (pentru a formata o dischetă în Linux, utilizați comanda <code>fdformat</code>)
<code>mlabel</code>	LABEL	Inițializează eticheta unui volum MS-DOS
<code>mmd</code>	MD sau MKDIR	Creează un director MS-DOS
<code>mrdd</code>	RD sau RMDIR	Șterge un director MS-DOS
<code>mread</code>	COPY	Copiază un fișier din MS-DOS în Linux
<code>mren</code>	REN sau RENAME	Redenumeste un fișier MS-DOS existent
<code>mtype</code>	TYPE	Afișează conținutul unui fișier MS-DOS
<code>mwrite</code>	COPY	Copiază un fișier din Linux în MS-DOS

Puteți lucra cu comenzile `mtools` exact în modul în care utilizați comenzile DOS corespunzătoare. De exemplu, comanda `mdir` operează exact ca și comanda `DIR` din DOS. Același principiu rămâne valabil și pentru celelalte comenzi `mtools` din tabelul 7-1. În ceea ce privește caracterele speciale (cum este spre exemplu `*`), amintiți-vă că primul program care interpretează comanda este chiar shellul Linux. Dacă nu doriți ca interpretorul să extindă un caracter special, trebuie să includeți toate numele fișierelor care conțin caractere speciale între ghilimele. Pentru a copia toate fișierele `*.txt` de pe unitatea A în directorul curent din Linux, utilizați de exemplu următoarea comandă:

```
mcopy "a:*.txt" .
```

Dacă nu puneți ghilimele, interpretorul va încerca să extindă șirul de caractere `a:*.txt` cu numele fișierelor din directorul curent, după care va încerca să copieze fișiere (dacă există) de pe discheta DOS.

Pe de altă parte, atunci când copiați fișiere din directorul curent Linux pe discheta DOS, este bine ca interpretorul să extindă toate caracterele speciale. Pentru a copia toate fișierele `*.pcx` din directorul curent din Linux pe o dischetă DOS, apăsați comanda `mcopy` în modul următor:

```
mcopy *.pcx a:
```

Utilitarele `mtools` vă permit să utilizați caracterul „backslash” (`\`) ca separator între numele directoarelor, întocmai ca în DOS. Numele fișierelor care conțin acest caracter trebuie incluse în ghilimele. Spre exemplu, comanda următoare realizează copierea unui fișier dintr-un subdirector aflat pe unitatea A în directorul curent din Linux:

```
mcopy "a:\test\sample.dat" .
```

Formatarea unei dischete DOS

Să presupunem că ați instalat sistemul Linux pe calculatorul PC de la domiciliu, pe care nu ați mai păstrat și sistemul MS-DOS, dar aveți nevoie să copiați câteva fișiere pe o dischetă MS-DOS pe care o veți utiliza la birou. Dacă dispuneți de o dischetă formatată în MS-DOS, puteți pur și simplu să montați acea dischetă și apoi să copiați fișierele din sistem pe dischetă utilizând comanda `cp` din Linux. Ce se întâmplă însă dacă nu aveți o dischetă formatată MS-DOS? Încă o dată, instrumentele `mtools` își dovedesc utilitatea.



Pachetul `mtools` dispune de utilitarul `mformat`, care poate formata o dischetă pe care o veți utiliza în MS-DOS. Spre deosebire de comanda `format` din DOS, care formatează o dischetă într-un singur pas, comanda `mformat` necesită parcurgerea unui proces în două etape pentru pregătirea unei dischete DOS:

1. Utilizați comanda `fdformat` (din Linux) pentru a formata fizic discheta. Comanda `fdformat` primește ca argument numele dispozitivului asociat unității de dischetă; acest nume conține toți parametrii necesari pentru formatarea dischetei.

Convenția de denumire a dispozitivelor asociate unităților de dischetă este ilustrată în figura 7-2. Pornind de la informațiile din figura 7-2, pentru formatarea unei dischete înaltă densitate (high-density) de 3,5 inch introduse în unitatea A a sistemului dumneavoastră, trebuie să utilizați comanda:

```
fdformat /dev/fd0H1440
```

```
Double-sided, 80 tracks, 18 sec/track. Total capacity is 1440 kB.
```

```
Formatting ... done
```

```
Verifying ... done
```

2. Utilizați comanda `mformat` pentru a instala un sistem de fișiere MS-DOS pe discheta formatată fizic. Dacă discheta este în unitatea A, tastați comanda:

```
mformat a
```

Utilizarea emulatorului DOSEMU

Accesul la sistemele de fișiere DOS prin intermediul comenzii `mount` sau al utilităților `mtools` vă permite să realizați un schimb de fișiere cu un sistem MS-DOS, facilitare care nu vă permite însă să și rulați programe DOS de sub Linux. După cum probabil vă așteptați, comunitatea Linux a găsit o soluție și la această problemă – puteți utiliza emulatorul DOSEMU pentru a crea în Linux un mediu de lucru similar cu MS-DOS.



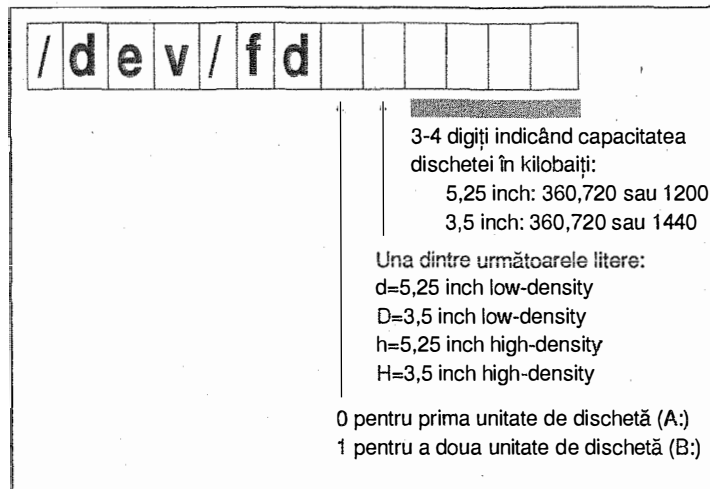
Secret

Chiar dacă numele DOSEMU provine de la *DOS emulator* (emulator de DOS), programul nu reprezintă în realitate un emulator. De fapt, DOSEMU exploatează aceleași facilități ale procesorului Intel 80x86 pe care la folosește și sistemul

Microsoft Windows atunci când rulează MS-DOS într-o fereastră. Începând cu procesorul 80386, procesoarele Intel 80x86 și Pentium pot lucra în mod virtual-86, în care mai multe mașini 8086 pot coexista pe un același procesor 80x86. După cum vă amintiți, 8086 este microprocesorul pe 16 biți care a rulat pentru prima dată MS-DOS. Datorită faptului că DOSEMU utilizează modul virtual-86, poate rula programe DOS în forma lor nativă, fără necesitatea de a emula setul de instrucțiuni al procesoarelor 80x86. Pentru a putea rula DOSEMU, aveți nevoie de o copie a sistemului MS-DOS (orice versiune a acestuia, de la 3.3 până la 6.2).

Figura 7-2

Convenția de denumire a unităților de dischetă în Linux



DOSEMU simulează un mediu DOS pentru aplicațiile DOS care doresc accesul direct la resurse ale sistemului cum sunt tastatura, imprimanta, porturile seriale, monitorul sau discul. Modul în care DOSEMU controlează diferitele resurse este stabilit în fișierul de configurare `/etc/dosemu.conf`.



Atenție!

Chiar dacă în urma lecturii acestei secțiuni veți cunoaște câte ceva despre DOSEMU, este bine să știți că acesta se află încă în curs de dezvoltare. Versiunea curentă a programului DOSEMU este 0.66 (acesta nu a fost considerat suficient de stabil pentru a se trece la versiunea 1.0). Chiar dacă DOSEMU este suficient de stabil, trebuie să fiți oricând pregătit de a reporni sistemul când apar anumite condiții critice. Salvați-vă cele mai importante fișiere din sistem înainte de a încerca să utilizați DOSEMU.

Instalarea DOSEMU

Programul de instalare al sistemului Red Hat Linux va instala și DOSEMU dacă, la instalarea sistemului Linux de pe CD-ROM-ul anexat acestei cărți, veți selecta componenta DOS/Windows Connectivity. Dacă ați omis să instalați DOSEMU, procedați după cum urmează pentru a instala DOSEMU imediat:

1. Dacă nu ați intrat în sistem ca root, tastați comanda `su` și apoi introduceți parola de root pentru a deveni root.
2. Asigurați-vă că în unitatea CD-ROM se găsește CD-ROM-ul anexat acestei cărți. Dacă CD-ROM-ul nu este deja montat, montați-l în sistem utilizând comanda:

```
mount /dev/cdrom /mnt/cdrom
```

3. Schimbați directorul curent de lucru cu directorul în care se găsesc pachetele Red Hat:

```
cd /mnt/cdrom/RedHat/RPMS
```

4. Tastați următoarea comandă pentru a instala DOSEMU:

```
rpm -i dosemu*
```

Toate fișierele necesare pentru rularea DOSEMU vor fi instalate în directoarele adecvate după terminarea acestei comenzi `rpm`.

Citirea manualului

După ce ați instalat DOSEMU, se recomandă să citiți documentația referitoare la configurarea și utilizarea DOSEMU. Pentru început, citiți fișierul text `QuickStart` din directorul `/usr/doc/dosemu*`. (Numele exact al acestui director depinde de versiunea DOSEMU pe care o utilizați.) Utilizați secvența de comenzi următoare pentru a vizualiza conținutul acestui fișier:

```
cd /usr/doc/dosemu*
less QuickStart
```



Sugestie

Comanda `less` vă permite să vizualizați un fișier pagină cu pagină. Comanda `less` este similară cu `more`, exceptând faptul că permite deplasarea atât înainte, cât și înapoi.

Iată câteva aspecte cheie explicate în fișierul `QuickStart`:

- Fișierul `/etc/dosemu.users` conține numele utilizatorilor care au dreptul de a rula DOSEMU, precum și operațiile pe care aceștia le pot executa în DOSEMU.
- Trebuie să editați fișierul de configurare `/etc/dosemu.conf` pentru a reflecta configurația sistemului dumneavoastră.
- Aveți nevoie de o *imagine a discului* (adică un fișier pe care DOSEMU îl tratează ca pe o unitate DOS). Acest fișier este prezent după instalarea DOSEMU de pe CD-ROM-ul anexat acestei cărți.
- Pentru a configura DOSEMU prima dată, vă este necesară o dischetă DOS care să conțină fișierele `FDISK.EXE` și `SYS.COM`. Puteți crea această dischetă sub DOS, sistem deja instalat pe PC-ul dumneavoastră.

În secțiunea următoare, veți învăța cum puteți utiliza o dischetă DOS pentru a configura o imagine a discului. (Acest fișier imagine va fi utilizat ca unitate C sub DOSEMU.)

Configurarea DOSEMU

Distribuția DOSEMU de pe CD-ROM-ul atașat lucrării este foarte aproape de a deveni complet funcțională. Trebuie însă să treceți prin câteva etape de configurare înainte de a putea rula DOSEMU. Aceste etape sunt descrise în secțiunile următoare.

Editarea fișierului `/etc/dosemu.conf`

Trebuie să lucrați ca root atunci când efectuați aceste operații de configurare. Prima dintre aceste operații este editarea fișierului `/etc/dosemu.conf`. Ca și majoritatea fișierelor de configurare din Linux, `/etc/dosemu.conf` este un fișier text. Liniile comentariu încep cu caracterul diez (#). Fișierul inițial conține deja multe linii comentariu, precum și instrucțiuni pentru fiecare secțiune în parte.

Pentru început, observați următoarele elemente în fișierul de configurare:

1. Căutați șirul de caractere `hdimage` și asigurați-vă că linia pe care se află nu este comentată, apărând sub forma următoare:

```
disk { image "/var/lib/dosemu/hdimage" } # use diskimage file.
```

Această linie se referă la fișierul `/var/lib/dosemu/hdimage`, care joacă rolul de disc sub DOSEMU. Numele de `hdimage` este o prescurtare de la *hard disk image* (imagina a hard-discului). Prima linie `disk` din fișierul `/etc/dosemu.conf` definește unitatea C de sub DOSEMU. La instalarea DOSEMU, programul de instalare creează inclusiv fișierul `/var/lib/dosemu/hdimage`, care va fi utilizat cu rolul de imagine a discului.

2. Dacă aveți o partiție DOS la care doriți să aveți acces de sub DOSEMU, eliminați caracterul diez de pe linia corespunzătoare (sau adăugați o linie nouă cu numele exact al partiției DOS). Dacă numele partiției DOS este `/dev/hda1`, în fișierul `/etc/dosemu.conf` următoarea linie trebuie să fie necomentată:

```
disk { partition "/dev/hda1" readonly } # 1st partition on 1st IDE.
```

Această partiție va deveni unitatea D sub DOSEMU.

3. Căutați șirul de caractere `fd0` și asigurați-vă că dispozitivele `/dev/fd0` (prima unitate de dischetă) și `/dev/fd1` (a doua unitate de dischetă) au dimensiunea corect specificată (*threeinch* sau *fiveinch*). Sistemul meu dispune de o singură unitate de 3,5 inch, prin urmare singura linie care nu este comentariu în secțiunea FLOPPY DISKS din fișierul `/etc/dosemu.conf` este următoarea:

```
floppy { device /dev/fd0 threeinch }
```

Fișierul `/etc/dosemu.conf` conține multe alte opțiuni, pe care deocamdată nu este necesar să le aducem în discuție. Înainte de toate, trebuie ca DOSEMU să funcționeze. Puteți apoi să activați anumite opțiuni specifice, cum ar fi imprimanta sau accesul la portul serial de sub DOSEMU.

Inițializarea imaginii discului

Imaginea discului poate fi gândită ca reprezentând un disc neformatat; acesta trebuie formatat și trebuie să permită încărcarea sistemului de operare înainte ca DOSEMU să poată fi lansat utilizând această imagine. Pentru aceasta, aveți nevoie de o dischetă de boot în MS-DOS. Puteți crea această dischetă utilizând sistemul MS-DOS instalat inițial pe PC-ul dumneavoastră. Pentru a crea o dischetă de pornire în DOS, trebuie mai întâi să porniți sistemul sub MS-DOS (sau Windows 95), parcurgând apoi următoarele etape:

1. Introduceți o dischetă în unitatea A (toată informația care se găsește pe dischetă va fi distrusă prin formatare) și tastați comanda **FORMAT A: /S**. Sistemul MS-DOS va răspunde cu următorul mesaj:

```
Insert new diskette in drive A:
and press ENTER when ready...
```

2. Apăsați Enter, pentru că deja ați introdus discheta în unitatea A. Sistemul MS-DOS va formata discheta, copiind totodată fișierele necesare încărcării sistemului de operare (această operație este determinată de opțiunea `/S` a comenzii **FORMAT**).
3. Sistemul MS-DOS vă cere să introduceți o etichetă de volum. Ca răspuns, apăsați Enter.
4. Sistemul DOS vă întreabă apoi dacă doriți să formatați o altă dischetă. Tastați `n`, pentru a arăta că ați încheiat operația de formatare.

5. Copiați fișierele **FDISK.EXE** și **SYS.COM** pe dischetă. Aceste două programe sunt necesare pentru inițializarea imaginii discului sub DOSEMU. Pentru a copia programele **FDISK.EXE** și **SYS.COM** pe dischetă, utilizați următoarele comenzi:

```
c:
cd \windows\command (de regula, fisierele MS-DOS se gasesc in acest director)
copy fdisk.exe a:
copy sys.com a:
```

După pregătirea dischetei de pornire în DOS, reporniți sistemul Linux în sistem, utilizând metoda preferată (LILO sau discheta de boot în Linux). Intrați apoi în sistem ca `root`.

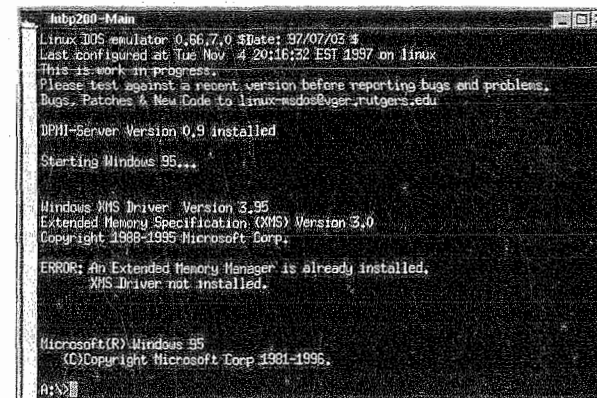
Introduceți discheta DOS în unitatea A și tastați următoarea comandă la promptul shellului:

```
dos -A
```

După o scurtă pauză, DOSEMU va porni de pe unitatea A și va apărea promptul `A:\>`, după cum se observă în figura 7-3.

Figura 7-3

Prima lansare a DOSEMU de pe o dischetă de boot în DOS



Tastați următoarea comandă DOS:

```
A:\> dir c:
```

DOSEMU va afișa conținutul unității C, adică al imaginii discului pe care DOSEMU o utilizează în locul unității C. Următoarea operație este de a activa această unitate C pentru pornire. Pentru inițializarea imaginii discului (unitatea C de sub DOSEMU), executați următoarea secvență de operații:

1. Pentru inițializarea înregistrării principale de boot (MBR) a unității C, tastați următoarea comandă:

```
A:\> fdisk /mbr
```

Vor apărea câteva mesaje de avertisment, pe care le puteți ignora.

2. Transferați apoi fișierele necesare ale sistemului pe imaginea discului, utilizând următoarea comandă:

```
A:\> sys c:
System transferred
```

După aceste două etape, imaginea discului (unitatea C de sub DOSEMU) este activată pentru pornire. Înainte de a testa această imagine, părăsiți emulatorul DOSEMU utilizând comanda:

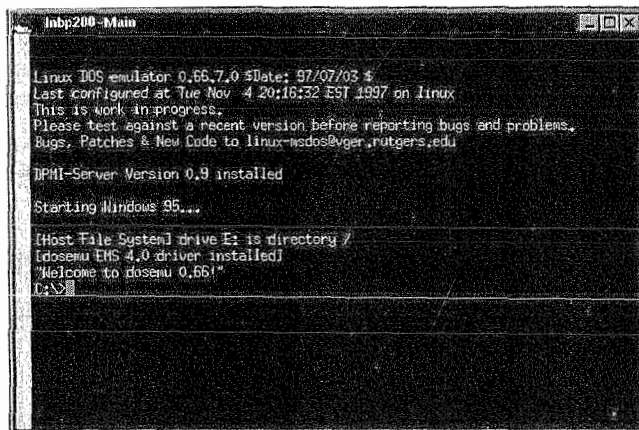
```
A:\> c:\exitemu
```

Lansarea DOSEMU de pe imaginea discului

Dacă ați configurat corect imaginea discului, trebuie să puteți porni DOSEMU chiar de pe acea imagine. Extrageți discheta DOS din unitatea A a sistemului și tastați `dos` la promptul interpretorului. DOSEMU va porni de pe imaginea discului, afișând promptul binecunoscut `C:\>`, după cum se observă și în figura 7-4.

Figura 7-4

Pornirea DOSEMU de pe imaginea discului



În figura 7-4 observați că DOSEMU afișează anumite informații, după care pornește de pe unitatea C. De la prompt-ul `C:\>`, veți putea introduce comenzi DOS.

Dacă veți tasta `dir`, veți observa că imaginea discului care ține locul unității C are aproximativ 1 MB. Această dimensiune este suficientă, deoarece, pe lângă partiția Linux, veți avea acces și la partiția DOS existentă. Unitatea C este, în acest caz, utilizată numai pentru pornirea sistemului MS-DOS.



Sugestie

Mesajul inițial afișat de DOSEMU indică litera care desemnează sistemul de fișiere Linux (directorul `/`). De exemplu, mesajul afișat în figura 7-4 ne anunță că directorul `/` este reprezentat prin unitatea E. Aceasta înseamnă că tastând `dir e:`, veți putea vedea conținutul directorului rădăcină (`/`) din Linux.

Dacă pe discul sistemului există o partiție DOS și ați adăugat o linie disk corespunzătoare acelei partiții în fișierul `/etc/dosemu.conf`, acea partiție va fi unitatea D sub DOSEMU. Executați comanda `dir d:` și comparați ceea ce afișează DOSEMU cu conținutul real al partiției DOS. Din DOSEMU puteți rula câteva aplicații DOS simple (cum ar fi editorul din MS-DOS) de pe partiția DOS a discului. Tastați `d:\windows\command\edit sample.txt` pentru a lansa editorul MS-DOS în vederea editării fișierului `sample.txt`. În figura 7-5 puteți observa editorul MS-DOS rulând într-o sesiune DOSEMU dintr-o fereastră `xterm`.

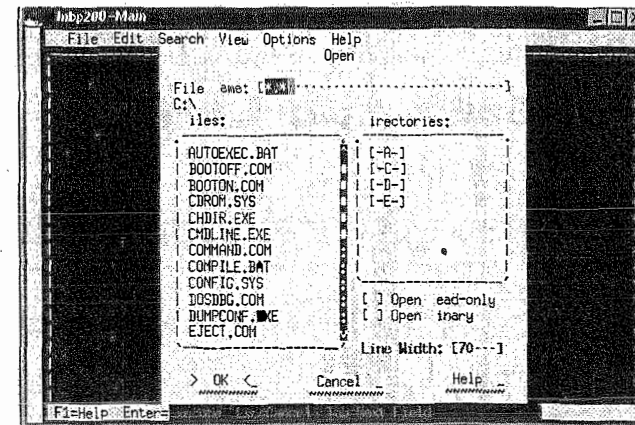
Figura 7-5 prezintă caseta de dialog Open a editorului (apăsăți `Alt+F` și apoi tastați `o` pentru a deschide această casetă). Pentru a părăsi editorul, apăsăți `Alt+F` și apoi tastați `x`.

Dacă veți lansa DOSEMU din mod text și nu dintr-o fereastră `xterm`, ecranul va avea un aspect similar cu al unui ecran DOS adevărat.

Dacă doriți să părăsiți DOSEMU, tastați comanda `c:\exitemu` la promptul DOS.

Figura 7-5

Caseta de dialog Open a editorului MS-DOS rulând sub DOSEMU într-o fereastră `xterm`



Rezumat

Dacă pe lângă partiția Linux, PC-ul dumneavoastră mai are și o partiție DOS sau dacă doriți să lucrați cu dischete DOS, puteți avea acces la fișierele DOS direct din Linux. Puteți de asemenea să utilizați utilitarele `mtools` pentru a formata și utiliza o dischetă DOS direct din Linux. În plus, un proiect aflat încă în lucru – DOSEMU – vă oferă un emulator al sistemului DOS, care vă permite să rulați anumite aplicații DOS direct din Linux.

După lectura acestui capitol, ați învățat următoarele:

- ▶ Sistemul Linux dispune de facilități încorporate de lucru cu sistemul de fișiere MS-DOS. Puteți utiliza comanda `mount` pentru a avea acces la o partiție DOS sau la o dischetă DOS din Linux. După montarea unui sistem de fișiere DOS într-un director din Linux, puteți utiliza comenzi cum sunt `ls` sau `cp` pentru manipularea directă a fișierelor DOS.
- ▶ La instalarea sistemului Linux, urmând instrucțiunile din capitolul 1, ați instalat de asemenea și un set de programe utilitare, cunoscut sub numele de `mtools`. Programele `mtools` vă oferă o modalitate convenabilă de acces la fișiere MS-DOS, în special la dischete, deoarece comenzile `mtools` nu necesită o montare în prealabil a dischetei DOS. Printre utilitarele `mtools` se află comenzi cum ar fi `mdir` sau `mcopy`, care funcționează similar cu comenzile DOS `DIR` și `COPY`.
- ▶ CD-ROM-ul anexat acestei cărți mai conține o versiune preliminară a emulatorului DOSEMU, care utilizează modul `virtual-86` al procesoarelor Intel 80x86 pentru a rula sistemul MS-DOS. DOSEMU poate fi configurat chiar la instalarea sistemului Linux. În caz contrar, instalarea DOSEMU se efectuează de pe CD-ROM, utilizând o comandă `rpm -i`. Pentru configurarea DOSEMU, aveți nevoie de o dischetă de boot în MS-DOS. Chiar dacă este numai o versiune preliminară, DOSEMU permite rularea unui mare număr de aplicații DOS existente.

Capitolul 8

Programarea sub Linux

în Tcl/Tk

În acest capitol:

- ▶ Veți examina Tcl/Tk
- ▶ Veți înțelege sintaxa limbajului Tcl
- ▶ Veți scrie scripturi Tcl
- ▶ Veți construi interfețe grafice pentru scripturile Tcl în Tk
- ▶ Veți învăța Tcl/Tk prin exemple

Dacă știți deja să programați în limbajul C și în X Window System, veți fi surprins de ușurința cu care puteți crea aplicații grafice utilizând limbajul Tool Command Language (Tcl) și setul de instrumente X asociat acestui limbaj, Tk – denumite în comun prin termenul Tcl/Tk. Tcl este un limbaj utilizat la elaborarea de scripturi, asemănător cu Perl. Cel mai important avantaj al limbajului Tcl îl reprezintă setul de instrumente pentru X, Tk, datorită căruia se pot dezvolta scripturi cu o interfață grafică pentru utilizator.

Atunci când utilizați pentru prima dată Tcl/Tk, veți fi plăcut surprins aflând cât de puține linii sunt necesare în acest limbaj pentru crearea unei interfețe grafice funcționale. Pentru un începător absolut, scripturile Tcl/Tk par complicate, dar dacă ați utilizat seturi de instrumente cum sunt Xt sau Motif pentru a dezvolta aplicații, veți aprecia nivelul înalt la care lucrează Tk. Crearea unei componente simple a interfeței cu utilizatorul (cum este un buton) este mult mai simplă în Tk față de Motif. Există desigur și alte aspecte, cum ar fi dispunerea componentelor interfeței grafice, dar rezultatele sunt mult mai rapid vizibile atunci când se utilizează Tk comparativ cu un program în C care utilizează biblioteca Motif.



Sugestie

Dacă nu ați scris niciodată un program, nu trebuie să evitați acest capitol de frică. Exemplele din acest capitol vă prezintă bazele limbajelor Tcl și Tk. Veți deveni cu siguranță un susținător al celor două limbaje după ce veți vedea cât de rapid puteți crea aplicații cu interfețe grafice utilizând interpretorul Tcl/Tk.

Originea limbajelor Tcl/Tk

Cele două limbaje (Tcl și Tk) au fost create de John Ousterhout, când studia la Universitatea din Berkeley, California. Limbajul Tcl a apărut pentru prima dată în 1989, fiind urmat de Tk în 1991. Tcl/Tk sunt ambele disponibile în mod gratuit pentru o utilizare

restricționată, inclusiv în scopuri comerciale. În momentul scrierii acestei lucrări, versiunea limbajului Tcl este 8.0; limbajul Tk a ajuns de asemenea tot la versiunea 8.0. Cele două limbaje se găsesc pe CD-ROM-ul anexat acestei cărți.

O introducere în limbajul Tcl

Creatorul limbajului Tcl, John Ousterhout, a dorit să creeze un limbaj de comandă simplu, al cărui interpretor să poată fi legat cu orice program C, pentru ca orice program C să poată include scripturi Tcl. Termenul *incorporabil*¹ desemnează posibilitatea ca orice program C să utilizeze interpretorul Tcl și să ruleze scripturi Tcl.

Următoarele câteva secțiuni conțin o trecere în revistă a limbajului Tcl și a sintaxei acestuia, și prezintă câteva dintre cele mai importante comenzi. Deoarece limbajul Tcl stă la baza setului de instrumente Tk, trebuie mai întâi să vă familiarizați cu acest limbaj înainte de a trece la Tk (Tk este fără îndoială mult mai spectaculos, datorită interfețelor grafice pe care le puteți crea).

Primul dumneavoastră script în Tcl

În capitolul 6 ați învățat să scrieți scripturi în shell și în Perl. Scripturile Tcl se elaborează în aceeași manieră. Spre deosebire de Perl, Tcl include un interpretor interactiv al comenzilor Tcl introduse. Numele interpretorului Tcl este `tclsh` și acesta se găsește de regulă în directorul `/usr/bin`.

La intrarea în sistem, variabila de mediu `PATH` include de regulă directorul `/usr/bin`. În consecință, puteți lansa shellul Tcl tastând direct `tclsh`. Pe linia imediat următoare, va apărea semnul procent (%); acest semn este promptul interpretorului Tcl. Pentru a determina versiunea Tcl de care dispuneți, tastați `info tclversion` la promptul interpretorului Tcl. Acesta va răspunde afișând versiunea limbajului Tcl:

```
tclsh
% info tclversion
8.0
%
```

Puteți acum să încercați să executați în mod interactiv programul Tcl următor, care afișează șirul de caractere `Hello, World!` la ieșirea standard (adică pe ecran):

```
% puts "Hello, World!"
Hello, World!
%
```

Tastați `exit` pentru a părăsi shellul Tcl (`tclsh`).



Observație

Deși în capitolele anterioare nu am prezentat promptul interpretorului, în capitolul de față vom face excepție de la această regulă. Aceasta deoarece promptul interpretorului Tcl este diferit față de cel din Bash. În acest mod, vă veți aduce aminte că nu lucrați în Bash.

Shellul Tcl prelucrează imediat o comandă introdusă și afișează rezultatele acesteia, după care afișează promptul, așteptând o nouă comandă. Deocamdată, puteți tasta `exit` pentru a părăsi interpretorul Tcl.

Pentru a pregăti și lansa un script Tcl, procedați după cum urmează:

1. Utilizați un editor de texte pentru a scrie următoarele linii într-un fișier numit `hellotcl` (acesta va fi chiar scriptul Tcl), salvând apoi fișierul:

```
#!/usr/bin/tclsh
# Un script simplu în Tcl
puts "Hello, World!"
```

2. Tastați următoarea comandă la promptul interpretorului, pentru a acorda fișierului `hellotcl` permisiunea de execuție (utilizând deci opțiunea `+x` a comenzii `chmod`):

¹ În limba engleză în original: *embeddable*. (n.red.)

```
chmod +x hello.tcl
```

3. Pentru a rula scriptul `hello.tcl`, tastați la promptul interpretorului:

```
hello.tcl
Hello, World!
```

Aceste etape trebuie parcurse pentru crearea și rularea oricărui script Tcl. În continuare, trebuie să învățați nuanțele sintaxei limbajului Tcl și multe alte reguli. Scopul secțiunii de față este de a vă oferi o introducere în limbajul Tcl.

Mai multe despre Tcl/Tk

Acest capitol este dedicat unei treceri în revistă a limbajelor Tcl și Tk, subliniind aspectele esențiale și prezentând un număr de exemple simple. Spațiul nu ne permite însă să cuprindem toată informația de care aveți nevoie pentru a utiliza eficient cele două limbaje. Datorită popularității celor două limbaje, dispuneți în prezent de un număr important de surse de informație care le sunt dedicate, atât sub formă tipărită (cărți), cât și pe diverse situri din Internet. Iată o listă scurtă a acestor resurse:

Cărți. Până în prezent, au apărut două lucrări importante dedicate limbajelor Tcl/Tk. Prima este *Tcl and the Tk Toolkit*, de John Ousterhout, creatorul celor două limbaje. Această lucrare descrie pe larg limbajele Tcl și Tk, explicând și modul de analiză a sirurilor de comandă Tcl. Cealaltă lucrare este *Practical Programming in Tcl and Tk, Second Edition* de Brent B. Welch. În această lucrare se găsesc mai multe exemple în Tcl și Tk. O a treia lucrare apărută recent este *Tcl/Tk for Dummies* (apărută la editura IDG Books Worldwide în 1997), de Tim Webster și Alex Francis.

Resurse pe Internet. Distribuțiile cele mai recente ale limbajelor Tcl/Tk, împreună cu o colecție de

informații referitoare la dezvoltarea limbajelor, pot fi găsite pe anumite situri FTP sau World Wide Web. În lista următoare, acestea sunt indicate utilizând sintaxa Uniform Resource Locator (URL). Puteți utiliza orice browser Web (cum ar fi Netscape, Mosaic sau Lynx), introducând URL-ul dorit. Browserul va afișa informația corespunzătoare conținută în cadrul sitului respectiv.

<ftp://ftp.smli.com/pub/tcl/> (Situl oficial de distribuție al limbajelor Tcl/Tk)

<http://www.scriptics.com/> (Situl noii firme înființate de John Ousterhout)

<http://www.tclconsortium.org/> (Consortiul Tcl/Tk)

<http://www.neosoft.com/tcl/> (Arhiva contribuțiilor la sursele Tcl)

Alte situri FTP care conțin distribuții Tcl/Tk sunt următoarele:

<ftp://ftp.ibp.fr/pub/tcl/distrib>

<ftp://sunsite.doc.ic.ac.uk/packages/tcl/>

ftp://ftp.luth.se/pub/unix/tcl/ftp.smli.com_mirror/

<ftp://ftp.funet.fi/pub/languages/tcl/>

O trecere în revistă a limbajului Tcl

Așa cum o arată și numele (Tool Command Language), Tcl constă dintr-un set de comenzi care pot fi combinate respectând un anumit set de reguli. Pentru a redacta scripturi în Tcl, trebuie să înțelegeți două aspecte de bază:

■ **Sintaxa limbajului Tcl.** Sintaxa Tcl este reprezentată de setul de reguli pe care interpretorul Tcl le respectă atunci când interpretează un *șir de comandă* (adică o linie care conține o comandă și argumentele acesteia).

■ **Comenzile Tcl.** Chiar dacă sintaxa este comună tuturor comenzilor, fiecare comandă Tcl în parte este destinată unei anumite activități. Pentru a utiliza eficient limbajul, trebuie să cunoașteți care sunt comenzile disponibile și operația pe care o execută fiecare comandă.

Setul comenzilor Tcl poate fi extins de către aplicații. De fapt, însuși setul de instrumente Tk este o extensie a limbajului Tcl; Tk adaugă comenzi care manipulează componente ale interfețelor grafice.

Începeți prin a învăța sintaxa limbajului Tcl, adică un număr de reguli care determină modul în care este interpretată fiecare comandă a limbajului. Cum în Tcl există un număr considerabil de comenzi, învățarea acestora este o activitate de durată. Chiar după ce dobândiți abilitate în ceea ce privește sintaxa limbajului și stăpâniți un număr de comenzi, va fi necesar să lucrați cu ajutorul unui manual de comenzi pentru a verifica formatul exact al parametrilor fiecărei comenzi în parte.

Comenzile limbajului Tcl pun la dispoziția programatorilor următoarele facilități de bază, ce pot fi regăsite în orice alt limbaj de programare:

■ **Variable** pentru memorarea datelor. Fiecare variabilă are un nume și o valoare. În Tcl puteți de asemenea defini tablouri de variabile.

■ **Expresii** care combină valorile variabilelor, utilizând operatori. De exemplu, pentru a aduna valorile a două variabile, puteți recurge la utilizarea unei expresii. În Tcl, expresiile sunt evaluate utilizând comanda `expr`.

■ **Comenzi de control al fluxului**, care permit modificarea ordinii de execuție a comenzilor, în funcție de valoarea unei anumite expresii. Controlul fluxului comenzilor în scripturile Tcl se realizează cu ajutorul unor comenzi specifice cum sunt `for`, `foreach`, `break`, `continue`, `if`, `while` sau `return`.

■ **Proceduri** care permit gruparea mai multor comenzi sub un anumit nume. Procedurile acceptă, de asemenea, parametri. Comanda Tcl care permite definirea de proceduri este `proc`. Atunci când aveți de executat de mai multe ori aceeași secvență de comenzi (de regulă, cu argumente diferite), este mult mai simplu să apelați procedura care reprezintă acea secvență de comenzi.

Următoarele secțiuni tratează la modul general sintaxa limbajului Tcl și cele mai importante comenzi Tcl.

Bazele sintaxei limbajului Tcl

Pentru a înțelege bazele sintaxei limbajului Tcl, trebuie să aveți un minim de cunoștințe privind modul în care interpretorul Tcl procesează fiecare șir de comandă. Etapele parcurse sunt următoarele:

1. Interpretorul Tcl analizează (desparte) șirul de comandă în cuvinte.
2. Interpretorul aplică reguli de substituție a numelor variabilelor cu valorile acestora și înlocuiește anumite comenzi cu rezultatele lor.
3. Interpretorul execută comenzile, considerând că primul cuvânt din șir este numele comenzii și apelând o procedură care va executa comanda. Această procedură va trata restul cuvintelor din linia de comandă ca șiruri de caractere.

Atunci când scrieți comenzi Tcl, trebuie să separați numele comenzii de argumentele acesteia prin spațiu sau tab. Sfârșitul unei comenzi este marcat printr-un caracter linie nouă sau prin punct și virgulă (;). Puteți insera două comenzi pe aceeași linie, dacă le separați prin punct și virgulă. Prin urmare, puteți scrie următoarea linie de comandă:

```
puts Hello, ; puts World!
Hello,
World!
```


Rezultatul apare despărțit pe două linii, deoarece comanda `puts` adaugă în mod prestabilit un caracter de trecere la linie nouă.

Utilizați un backslash (\) la sfârșitul unei linii pentru a putea continua un șir de comandă pe linia următoare. În consecință, puteți rescrie șirul de comandă pentru afișarea mesajului `Hello, World!` în modul următor:

```
puts "Hello, \
World!"
```

Substituții

Interpretorul Tcl efectuează înlocuirea unor anumite părți ale șirului de comandă cu valori echivalente. De exemplu, dacă în fața numelui unei variabile apare semnul dolar (\$), interpretorul va înlocui numele variabilei cu valoarea acesteia. După cum veți învăța în secțiunea „Variabile”, pentru a defini o variabilă într-un script Tcl puteți utiliza comanda `set`, după cum urmează:

```
set count 100
```

Această comandă definește o variabilă numită `count` care are valoarea 100. Să presupunem că acum introduceți următoarea comandă:

```
puts $count
```

Interpretorul va înlocui mai întâi șirul `$count` cu valoarea variabilei, care este 100. Prin urmare, șirul de comandă va deveni:

```
puts 100
```

Interpretorul va executa această comandă, afișând valoarea 100. Acesta este un exemplu simplu de *substituție de variabile*.

În total, interpretorul limbajului Tcl permite trei tipuri de substituții:

- **Substituția variabilelor.** După cum ați observat din exemplul anterior, dacă interpretorul găsește semnul dolar (\$), înlocuiește acest semn și numele variabilei următoare cu valoarea corespunzătoare.

- **Substituția secvențelor „backslash”.** Puteți îngloba caractere speciale, cum sunt linie nouă și caracterul de tabulare, într-un anumit cuvânt, utilizând acest tip de substituție. Este suficient să tastați caracterul backslash, urmat de unul sau mai multe caractere; interpretorul va înlocui această secvență cu un caracter neafișabil. Aceste secvențe corespund cu secvențele escape din standardul ANSI C. În tabelul 8-1 puteți găsi secvențele backslash pe care interpretorul limbajului Tcl le înțelege.

- **Substituția comenzilor.** Acest tip de substituție se referă la mecanismul prin care puteți determina ca o anumită parte a șirului de comandă să fie evaluată și apoi înlocuită prin rezultatul întors, înainte ca interpretorul să prelucreză întregul șir de comandă. De exemplu, comanda `string length` „Hello, World!” returnează valoarea 13, care reprezintă lungimea șirului de caractere. Pentru a atribui unei variabile numite `len` lungimea acestui șir, tastați următoarea comandă:

```
set len [string length "Hello, World!"]
```

Interpretorul va evalua comanda cuprinsă în interiorul parantezelor pătrate, înlocuind apoi partea corespunzătoare a șirului de comandă cu valoarea comenzii evaluate. Prin urmare, șirul de comandă va deveni:

```
set len 13
```

comanda `set` atribuind variabilei `len` valoarea 13.

Tabelul 8-1 Secvențe backslash în limbajul Tcl

Secvență	Caracter de înlocuire*
\a	Caracterul bell (0x7)
\b	Backspace (0x8)
\f	Salt de pagină (0xc)
\n	Linie nouă (0xa)
\r	Retur de car (0xd)
\t	Tab orizontal (0x9)
\v	Tab vertical (0xb)
\<newline>	Caracterul linie nouă și spațiul de pe linia următoare sunt înlocuite cu un singur spațiu
\\	Un singur backslash (\)
\"	Ghilimele duble (")
\ooo	Valoarea specificată (de cel mult trei cifre) este interpretată în octal
\xhh	Valoarea specificată (cu cel mult două cifre) este interpretată în hexazecimal

* În paranteze, se găsesc valorile hexazecimale asociate diferitelor caractere speciale.

Comentarii

Începutul unui comentariu este marcat prin semnul „diez” (#); interpretorul limbajului Tcl va ignora restul liniei, începând cu acest semn. În Tcl, există totuși o restricție asupra comentariilor: acestea nu pot începe în interiorul unei comenzi. Înainte de a începe un comentariu, trebuie ca șirul curent de comandă să se termine.

Pentru a înțelege mai bine această problemă, încercați să executați următoarea comandă la prompt-ul interpretorului tclsh:

```
% puts "Hello, World" # Acesta este un comentariu
wrong #args: should be "puts" ?-newline? ?field? string
```

Comanda `puts` va parcurge restul liniei, semnalând o eroare în ceea ce privește numărul de argumente. Soluția este de a pune caracterul punct și virgulă înainte de diez, ca în exemplul:

```
% puts "Hello, World"; # Acesta este un comentariu
Hello, World!
```



Sugestie

Atunci când doriți să inserați un comentariu la sfârșitul unei comenzi Tcl, nu uitați ca înainte de semnul diez să puneți caracterul punct și virgulă. Acest caracter indică sfârșitul șirului de comandă și vă permite să începeți un comentariu.

Acolade și ghilimele

Pentru gruparea mai multor cuvinte, puteți utiliza acolade ({...}) sau ghilimelele duble ("..."). Utilizați ghilimelele pentru transferul argumentelor care conțin spații sau caractere punct și virgulă, utilizate în mod obișnuit cu rol de caractere de terminare a comenzii. Ghilimelele nu fac parte din comandă, fiind folosite numai la marcarea începutului, respectiv sfârșitului, unui grup de cuvinte. Iată câteva exemple în care se utilizează ghilimele pentru gruparea cuvintelor:

```
% puts "Hello, World!"
Hello, World!
```

```
% puts "Introduceti 1; in caz contrar, fisierul nu va fi salvat!"
Introduceti 1; in caz contrar, fisierul nu va fi salvat!
```

La gruparea unor cuvinte cu ajutorul ghilimelelor, se pot efectua toate tipurile de substituții, după cum puteți observa în exemplul următor:

```
% puts "In sirul 'hello' sunt [string length hello] caractere";
In sirul 'hello' sunt 5 caractere
```

Interpretorul Tcl va înlocui subșirul cuprins între paranteze cu rezultatul comenzii `string length hello`, care returnează numărul de caractere din șirul `hello`, adică 5.

O altă posibilitate de a grupa cuvintele este prin intermediul acoladelor. Atunci când includeți cuvintele între acolade, interpretorul limbajului Tcl nu efectuează nici una dintre substituțiile cunoscute (dacă veți include cuvintele între ghilimele, interpretorul va efectua substituții). Să considerăm exemplul anterior, în care ghilimelele sunt înlocuite cu acolade:

```
% puts {In sirul 'hello' sunt [string length hello] caractere};
In sirul 'hello' sunt [string length hello] caractere
```

După cum puteți observa, interpretorul Tcl va transmite totul nemodificat, ca un singur argument.



Sugestie

Utilizați acoladele ca mecanism de grupare atunci când trebuie să transmiteți anumite expresii unor comenzi de control, cum sunt ciclurile `while` și `for`, sau unor proceduri.

Variabile

În Tcl, orice entitate este, în mod prestabilit, un șir de caractere. Numele variabilelor, precum și valorile acestora, sunt memorate ca șiruri de caractere. Pentru definirea unei variabile, utilizați comanda `set`. Următoarele comenzi, spre exemplu, definesc variabila `carte` cu valoarea "Secrete Linux, editia a doua", variabila `an` cu valoarea 1999 și variabila `pret` cu valoarea \$49.99:

```
set carte "Secrete Linux, editia a doua"
set an 1999
set pret $49.99
```

Pentru a face referință la valoarea unei variabile, precedați numele acesteia cu semnul „dolar” (\$). În consecință, pentru a afișa valoarea variabilei `carte`, vom utiliza comanda:

```
% puts $carte
Secrete Linux, editia a doua
```

Dacă veți utiliza comanda `set` cu un singur argument, aceasta va returna valoarea argumentului respectiv. Comanda `set carte` este deci echivalentă cu `$carte`, după cum puteți vedea și din exemplul următor:

```
% puts [set carte]
Secrete Linux, editia a doua
```

Expresii

Expresiile sunt generate prin combinarea variabilelor utilizând operatori matematici, care sunt + (adunare), - (scădere), * (înmulțire) și / (împărțire). Iată în continuare câteva exemple de expresii:

```
set count 1
$count + 1
$count + 5 - 2
2 + 3.5
```

În cadrul expresiilor, se pot utiliza atât nume de variabile, cât și numere. Pentru a facilita citirea expresiilor, utilizați o spațiere corespunzătoare. Pentru a stabili ordinea de evaluare a expresiilor, recurgeți la paranteze.

Pe lângă operatorii aritmetici de bază, în Tcl sunt incluse și anumite funcții matematice, cum ar fi `sin`, `cos`, `tan`, `log` și `sqrt`. Aceste funcții pot fi apelate exact la fel ca și în C, cu argumentele în paranteze, ca în exemplul următor:

```
set angle 1.5
2*sin($angle)
```

Puteți de asemenea să utilizați operatori logici, cum ar fi ! (negația), && (conjuncția) și || (disjuncția). Aveți de asemenea la dispoziție o mulțime de operatori de comparație – cum ar fi < (mai mic), > (mai mare), <= (mai mic sau egal), == (egal) și != (diferit de). Expresiile care utilizează operatori de comparație sau operatori logici au valoarea 1, dacă sunt adevărate, respectiv 0, dacă sunt false. Două expresii corect formate sunt următoarele:

```
count == 10
angle < 3.1415
```

Expresiile nu reprezintă comenzi propriu-zise. Acestea pot fi însă transmise ca argumente, dar numai către comenzi care acceptă expresii ca argumente. Două exemple binecunoscute de comenzi care necesită expresii ca argumente sunt `if` și `while`.

În Tcl, o expresie poate fi evaluată utilizând comanda `expr`. Exemplul următor vă arată cum puteți evalua o expresie în interiorul unei comenzi Tcl:

```
% set angle 1.5
1.5
% puts "Rezultat = [expr 2*sin($angle)]"
Rezultat = 1.99499
```

Chiar dacă în Tcl toate valorile sunt memorate ca șiruri de caractere, trebuie să utilizați numere acolo unde acestea sunt necesare. Dacă variabila `carte` a fost definită cu valoarea "Secrete Linux, editia a doua", nu puteți forma expresia `$carte + 1`, deoarece aceasta nu are sens.

Comenzi de control al fluxului

Comenzile de control al fluxului din limbajul Tcl vă permit să precizați ordinea în care interpretorul limbajului va executa comenzile. Puteți testa valoarea unei expresii cu ajutorul comenzii `if`, iar în caz că valoarea este adevărată (adică nenulă) să executați un anumit set de comenzi. În Tcl aveți la dispoziție comenzi de control al fluxului similare cu cele din C, cum ar fi `if`, `for`, `while` și `switch`. În secțiunea de față vom prezenta pe scurt aceste comenzi.



Orice comandă de control în Tcl este compusă dintr-un bloc de comandă (un grup de comenzi) pe care comanda îl execută după evaluarea unei expresii. Pentru a evita substituțiile (ca de exemplu înlocuirea variabilelor prin valorile acestora), se recomandă să includeți întregul bloc de comandă între acolade. Următoarele comenzi `if-else` vă prezintă un mod eficient de utilizare a acoladelor:

```
if { expresie } {
# Comenzi care se executa cand expresia este adevarata
  comanda_1
  comanda_2
} else {
# Comenzi care se executa cand expresia este falsa
#...
}
```

Acest mod de utilizare a acoladelor trebuie respectat cu strictețe în script-urile Tcl. În mod special, nu uitați să lăsați un spațiu între numele comenzii de control (cum ar fi `if`) și acolada de deschidere (`{`) care urmează după acest nume.

Comanda if

În forma sa cea mai simplă, comanda `if` din Tcl realizează evaluarea unei expresii executând un set de comenzi dacă acea expresie este adevărată (diferită de zero). În exemplul următor, se realizează comparația dintre valoarea unei variabile și un prag dat:

```
if { $errorCount > 25 } {
    puts "Prea multe erori!"
}
```

Puteți adăuga la această formă și o clauză `else`, pentru cazul în care expresia este evaluată la zero (fals). Iată un exemplu:

```
if { $user == "root" } {
    puts "Configurarea sistemului a fost lansata..."
} else {
    puts "Ne pare rau, trebuie sa rulati acest program ca \"root!\""
}
```

Comanda `if` din Tcl poate fi urmată de zero sau mai multe comenzi `elseif`, necesare pentru efectuarea unor teste ceva mai complexe, ca în exemplul următor:

```
puts -nonewline "Introduceti numarul versiunii: " ; # cerem
# utilizatorului o valoare
set version [gets stdin]; # citim numarul versiunii

if { $version >= 10 } {
    puts "Nu este necesara o actualizare"
} elseif { $version >= 6 && $version <= 9 } {
    puts "Efectuati actualizarea standard"
} elseif { $version >= 3 && $version < 6 } {
    puts "Trebuie sa reinstalati complet sistemul"
} else {
    puts "Ne pare rau, nu putem efectua actualizarea"
}
```

Comanda while

Comanda `while` execută un bloc de comenzi până când o anumită expresie devine falsă. De exemplu, ciclul `while` următor citește linii de la intrarea standard până când utilizatorul apasă secvența `Ctrl+D`:

```
while { [gets stdin line] != -1 } {
    puts $line
# Alte prelucrari asupra valorii $line
}
```

Chiar dacă sintaxa comenzii `while` pare simplă, trebuie să înțelegeți că între acolade se găsesc de fapt două argumente. Primul este chiar expresia, cel de-al doilea conține comenzile Tcl care vor fi executate dacă expresia este adevărată. Pentru a delimita ambele argumente, trebuie să utilizați întotdeauna acolade. Acestea au rolul de a împiedica interpretorul limbajului Tcl să evalueze conținutul; ceea ce se află în interiorul acoladelor va fi interpretat de comanda `while`.

Atunci când într-un ciclu `while` utilizați o variabilă cu rol de contor, puteți utiliza comanda `incr` pentru a incrementa acea variabilă. Puteți „sări” direct la sfârșitul unui ciclu utilizând

comanda `continue`; comanda `break` determină ieșirea din ciclu. Script-ul Tcl următor utilizează un ciclu `while` pentru a aduna numerele de la 1 la 10, cu excepția lui 5:

```
#!/usr/bin/tclsh
set i 0
set sum 0

while { 1 } {
    incr i ; # incrementam i
    if { $i == 5 } { continue } ; # daca i este 5, nu adunam
# valoarea
    if { $i > 10 } { break } ; # daca i depaseste 10, se
# iese din ciclu
    set sum [expr $sum+$i] ; # altfel, adauga i la sum
}
puts "Sum = $sum";
```

Rezultatul afișat la rularea acestui script trebuie să fie următorul:

```
Sum = 50
```

Comanda for

Comanda `for` are în limbajul Tcl patru argumente și trebuie scrisă în modul următor:

```
for {expr_1} {expr_2} {expr_3} {
    comenzi
}
```

Comanda `for` evaluează expresia `expr_1` o singură dată, la începutul ciclului, executând apoi comenzile cuprinse în ultima pereche de acolade până când valoarea expresiei `expr_2` devine zero. Comanda `for` evaluează expresia `expr_3` după fiecare execuție a comenzilor din corpul ciclului. Oricare dintre cele trei expresii poate lipsi, dar nu trebuie să omiteți acoladele. În exemplul următor, am utilizat un ciclu `for` pentru a aduna numerele de la 1 la 10:

```
#!/usr/bin/tclsh
for {set i 0; set sum 0} { $i <= 10 } {set sum [expr $sum+$i]; incr i} {
    puts "Sum = $sum";
}
```

Rezultatul afișat de acest script va fi:

```
Sum = 55
```

Comanda foreach

Această comandă nu are un echivalent direct în limbajul C, fiind însă extrem de utilă atunci când doriți să executați o anumită operație asupra fiecărei valori dintr-o listă de variabile. În exemplul următor, se adună o listă de numere date utilizând o comandă `foreach`:

```
set sum 0
foreach i { 1 2 3 4 5 6 7 8 9 10 } {
    set sum [expr $sum+$i]
}
puts "Sum = $sum"
```

Dacă o listă este memorată într-o variabilă, puteți utiliza numele acelei variabile în locul listei cuprinse în prima pereche de paranteze. În exemplul următor, ciclul `foreach` afișează șirurile de caractere dintr-o listă dată:

```
set users "root naba"
foreach user $users {
    puts "$user"
}
```

Comanda switch

Comanda switch din Tcl diferă semnificativ de instrucțiunea switch din C. În locul evaluării unei expresii aritmetice, comanda switch din Tcl compară un șir de caractere cu un set de modele, executând un anumit set de comenzi, în funcție de modelul care se potrivește cu șirul. Modelul este de obicei exprimat sub formă de expresie regulată.



Referință
Încrucișată

În capitolul 6, puteți găsi o scurtă prezentare a expresiilor regulate.

Sintaxa și modul obișnuit de utilizare a comenzii switch sunt ilustrate prin următorul exemplu:

```
#!/usr/bin/tclsh
# Acest script citește comenzile introduse de utilizator
# și le prelucrează utilizând o comanda switch
```

```
set prompt "Introduceți comanda (\quit\") pentru ieșire): "
```

```
puts -nonewline "$prompt"; flush stdout
```

```
while { [gets stdin cmd] != -1 } {
    switch -exact -- $cmd {
        quit { puts "Bye!"; exit }
        start { puts "Started" }
        stop { puts "Stopped" }
        draw { puts "Draw..." }
        default { puts "Comanda necunoscută: $cmd" }
    }
    # interoghează din nou utilizatorul
    puts -nonewline "$prompt"; flush stdout
}
```

Prezentăm în continuare un exemplu de sesiune de lucru cu acest script (comenzile introduse de utilizator sunt scrise cu caractere albine):

```
Introduceți comanda (\quit\") pentru ieșire): help
Comanda necunoscută: help
Introduceți comanda (\quit\") pentru ieșire): start
Started
Introduceți comanda (\quit\") pentru ieșire): stop
Stopped
Introduceți comanda (\quit\") pentru ieșire): quit
Bye!
```

După cum puteți deduce din acest exemplu, comanda switch compară un șir de caractere cu o mulțime de modele, după care execută un set de comenzi, în funcție de modelul cu care a detectat echivalența. În exemplul de mai sus, șirul de caractere comparat este \$cmd (care este inițializat prin citirea comenzilor introduse de utilizator folosind o comandă gets), iar șabloanele sunt șirurile quit, start, stop și draw. Indicatorul -exact din forma generală a comenzii switch forțează detectarea echivalenței exacte:

```
switch -exact -- $cmd {
    ...
}
```



Cele două liniuțe (--) de după opțiunea -exact marchează sfârșitul opțiunilor comenzii switch. Nu omiteți niciodată aceste două liniuțe de la sfârșitul secțiunii de comenzi, pentru a împiedica echivalențele accidentale dintre șirul de test și un indicator (flag).

Pentru compararea șirului de test cu una sau mai multe expresii regulate, utilizați comanda switch cu opțiunea -regexp, ca în exemplul următor:

```
# Presupunem ca șirul de test este $cmd
switch -regexp -- $cmd {
    ^q.* { puts "Bye!"; exit }
    ^x.* { puts "Opțiunea x..." }
    ^y.* { puts "Opțiunea y..." }
    ^z.* { puts "Opțiunea z..." }
    default { puts "Comanda necunoscută: $cmd" }
}
```

În acest exemplu, toate expresiile regulate au aceeași formă. Modelul ^z.* desemnează toate șirurile care încep cu o singură literă z, urmată de un număr nedefinit de alte caractere.

Proceduri în Tcl

Puteți utiliza comanda proc pentru a vă crea propriile dumneavoastră comenzi. Aceste comenzi sunt denumite *proceduri*; interpretorul limbajului Tcl le tratează nediferențiat față de comenzile Tcl native. Scrierea unei proceduri în Tcl este relativ simplă, după cum reiese și din următorul exemplu:

```
#!/usr/bin/tclsh

proc total items {
    set sum 0
    foreach i $items {
        set sum [expr $sum+$i]
    }
    return $sum
}

set counts "5 4 3 5"
puts "Total = [total $counts]"
```

La rulare, acest script va afișa următorul rezultat:

```
Total = 17
```

În acest exemplu, numele procedurii este total; aceasta primește ca argument o listă de numere. Procedura citește lista în variabila numită items. Corpul procedurii extrage fiecare element din listă și returnează suma tuturor elementelor. Prin urmare, pentru a aduna numerele de la 1 la 10, puteți apela procedura total ca în exemplul următor:

```
set sum1_10 [total {1 2 3 4 5 6 7 8 9 10}]
```



Secret

Într-o procedură Tcl, argumentul args are o semnificație specială: utilizând acest argument puteți trimite procedurii un număr variabil de argumente. Redenumind argumentul items al procedurii total cu numele args, veți putea ulterior apela procedura total în modul următor:

```
set sum1_10 [total 1 2 3 4 5 6 7 8 9 10]; # remarcăți numărul variabil de argumente
```

Pentru a avea acces la o *variabilă globală* (definită în afara procedurii) în procedura Tcl, trebuie să utilizați comanda global în interiorul procedurii. Această comandă determină ca o variabilă globală să devină vizibilă în interiorul domeniului de valabilitate al unei proceduri. Spre exemplu, dacă zona curentă de desenare dintr-un program Tk (setul de instrumente X din Tcl) este memorată în variabila theCanvas, pentru a utiliza aceeași zonă într-o procedură, în corpul procedurii trebuie să includem comanda:

```
global theCanvas
```

Comenzi Tcl înglobate

În exemplele anterioare, ați făcut deja cunoștință cu o mulțime de comenzi din limbajul Tcl. Cunoașterea tipurilor de comenzi disponibile vă ajută să decideți comenzile cele mai indicate pentru fiecare situație în parte. Chiar dacă în acest capitol nu putem prezenta toate comenzile Tcl, în tabelul 8-2 se găsesc principalele comenzi predefinite în Tcl.



Sugestie

Pentru a obține informații online în legătură cu o comandă Tcl din tabelul 8-2, tastează **man n** și numele comenzii. De exemplu, tastează **man n file** pentru informații privind comanda file.

Tabelul 8-2 Comenzi Tcl predefinite

Comanda	Acțiunea
append	Adaugă un argument la valoarea unei variabile.
array	Efectuează diferite operații asupra tablourilor.
break	Părăsește o comandă repetitivă (cum ar fi while sau for).
catch	Execută un script și detectează erorile pentru a împiedica propagarea acestora la interpretorul limbajului Tcl.
cd	Schimbă directorul curent de lucru.
close	Închide un fișier deschis.
concat	Concatenează două sau mai multe liste într-una singură.
continue	Trece imediat la iterația următoare a unui ciclu while sau for.
eof	Detectează sfârșitul de fișier pentru un fișier deschis.
error	Generează o eroare.
eval	Concatenează liste (la fel ca și concat), evaluând lista rezultată într-un script Tcl.
exec	Lansează unul sau mai multe procese care execută argumentele comenzii.
exit	Determină sfârșitul scriptului Tcl.
expr	Evaluează o expresie.
file	Verifică numele și atributele unui fișier.
flush	Scrie imediat ieșirea din buffer într-un fișier.
for	Implementează un ciclu for.
foreach	Execută o acțiune specificată asupra fiecărui element dintr-o listă.
format	Formatează ieșirea, pe care o memorează într-un șir de caractere (la fel ca și funcția sprintf din C).
gets	Citește o linie dintr-un fișier.
glob	Returnează numele fișierelor care se potrivesc cu un model dat (cum ar fi *.tcl)
global	Permite accesul la variabilele globale.
history	Permite accesul la lista de istoric (lista ultimelor comenzi Tcl).

Comanda	Acțiunea
if	Testează o expresie, executând un set de comenzi atunci când expresia este adevărată (adică diferită de zero).
incr	Incrementează valoarea unei variabile.
info	Returnează informațiile referitoare la interpretorul Tcl.
join	Creează un șir de caractere pr'conținutul tuturor elementelor dintr-o listă.
lappend	Adaugă elemente la sfârșitul unei liste.
lindex	Returnează elementul cu indexul specificat dintr-o anumită listă. (Primul element are indexul 0.)
linsert	Inserează elemente într-o listă înaintea unui index precizat.
list	Creează o listă compusă din parametrii specificați.
llength	Returnează numărul elementelor unei liste.
lrange	Returnează un domeniu specificat de elemente adiacente dintr-o listă.
lreplace	Înlocuiește elemente dintr-o listă cu alte elemente.
lsearch	Caută un anumit element dintr-o listă.
lsort	Sortează o listă într-o ordine precizată.
open	Deschide un fișier și returnează identificatorul obținut.
pid	Returnează identificatorul unui proces (PID)
proc	Definește o procedură Tcl.
puts	Scrie un șir de caractere într-un fișier.
pwd	Returnează directorul curent de lucru.
read	Citește un număr specificat de baiți dintr-un fișier. (Este posibil să citiți întregul fișier utilizând o singură instrucțiune read.)
regexp	Testează echivalența dintre un șir de caractere și o expresie regulată.
regsub	Substituie un model de expresie regulată cu altul.
rename	Redenumeste sau șterge o comandă.
return	Returnează o valoare dintr-o procedură Tcl.
scan	Analizează un șir de caractere, utilizând aceiași specificatori de format ca și funcția sscanf din C.
seek	Modifică poziția de acces (la care va avea loc următoarea operație de citire sau de scriere) într-un fișier deschis.
set	Modifică valoarea unei variabile sau returnează valoarea curentă a acesteia.
source	Citește un fișier, prelucrându-l ca pe orice alt script Tcl.
split	Desparte un șir de caractere într-o listă Tcl.
string	Efectuează diferite operații asupra șirurilor de caractere.
switch	Prelucrează un bloc de comenzi din mai multe posibilități, în funcție de potrivirea dintre un șir specificat și un model dintr-o mulțime de modele.

(continuare)

Tabelul 8-2 Continuare

Comanda	Acțiunea
tell	Returnează poziția curentă de acces într-un fișier deschis.
time	Returnează timpul total necesar pentru execuția unui script.
trace	Execută un set specificat de comenzi Tcl la fiecare acces asupra unei variabile.
unknown	Procesează comenzile necunoscute. (Interpretorul limbajului Tcl apelează această comandă ori de câte ori întâlnește o comandă pe care nu o cunoaște.)
unset	Elimină definiția uneia sau mai multor variabile.
uplevel	Execută un script într-un context diferit.
upvar	Referă o variabilă din afara unei proceduri. (Utilizată pentru implementarea transferului prin referință al argumentelor unei proceduri, în care modificarea unui argument în procedură determină modificarea copieii originale a celui argument)
while	Implementează un ciclu while, care execută repetat un set de comenzi Tcl atât timp cât valoarea unei expresii este nenulă (adică adevărată).

Lucrul cu șiruri de caractere în Tcl

Printre comenzile Tcl din tabelul 8-2, veți găsi câteva – cum ar fi `append`, `join`, `split`, `string`, `regexp` și `regsub` – care operează asupra șirurilor de caractere. Această secțiune prezintă câteva comenzi de lucru cu șiruri de caractere.

Atunci când atribuiți unei variabile un șir de caractere, interpretorul limbajului Tcl consideră acel șir de caractere ca pe o entitate de sine stătătoare, chiar dacă include spații sau caractere speciale. În unele situații, trebuie să despărțiți un șir de caractere într-o listă de elemente. Separarea unui șir de caractere în componente se realizează ușor utilizând comanda `split`. Spre exemplu, liniile din fișierul `/etc/passwd` sunt de forma următoare:

```
Root:NpYzwIedcGA0.:0:0:root:/root:/bin/bash
```

Această linie este compusă din mai multe câmpuri separate prin caracterul „două puncte” (:). Să presupunem că doriți să extrageți primul câmp de pe fiecare linie (acel câmp conținând numele utilizatorului în sistem). Fișierul este citit linie cu linie, după care fiecare linie este despărțită într-o listă, și se extrage primul element (cel cu indexul 0) din fiecare listă. Prezentăm în continuare scriptul Tcl care realizează această operație:

```
#!/usr/bin/tclsh
```

```
set fid [open "/etc/passwd" r] ;# deschidem fisierul de parole numai pentru citire
```

```
while { [gets $fid line] != -1 } {
    set fields [split $line ":"] ;# aceasta comanda desparte sirul line intr-o lista
    # Tipareste primul camp
    puts [lindex $fields 0] ; # lindex extrage elementul cu un index precizat
}
```

În urma rulării, acest script va afișa numele tuturor utilizatorilor din fișierul `/etc/passwd`.

Comanda `join` realizează operația inversă lui `split`; această comandă creează un singur șir de caractere prin concatenarea elementelor unei liste.

Să presupunem că avem o listă cu șase elemente, definită după cum urmează:

```
set x {1 2 3 4 5 6}
```

La concatenarea elementelor dintr-o listă, puteți selecta caracterul prin care vor fi despărțite câmpurile. De exemplu, pentru a concatena elementele din listă fără a introduce nici un caracter între ele, utilizați următoarea formă:

```
set y [join $x ""]
```

În urma execuției acestei comenzi, șirul de caractere `y` va avea valoarea `"123456"`.

Comanda `string` reprezintă de fapt un grup de comenzi care lucrează cu șiruri de caractere, primul argument al comenzii precizând operația care va fi executată. De exemplu, comanda `string compare` realizează comparația a două șiruri de caractere, returnând zero atunci când cele două șiruri sunt identice. Dacă valoarea întoarsă este -1, atunci primul parametru este mai mic (în ordine lexicală) decât al doilea, adică apare în dicționar înaintea celui de-al doilea. Reciproc, dacă funcția returnează valoarea 1, înseamnă că primul șir este lexical mai mare decât cel de-al doilea. Puteți deci să utilizați comanda `string compare` în cadrul unei comenzi `if`, după cum urmează:

```
if { [string compare $command "quit"] == 0 } {
    puts "Exiting..."
    exit 0
}
```

Operațiile care pot fi efectuate cu ajutorul comenzii `string` din Tcl se găsesc în tabelul 8-3.

Tabelul 8-3 Operații asupra șirurilor de caractere în Tcl

Comanda	Descriere
<code>string compare string1 string2</code>	Returnează -1, 0 sau 1 după compararea a două șiruri de caractere.
<code>string first string1 string2</code>	Returnează indexul primei apariții a șirului <code>string1</code> în cadrul lui <code>string2</code> .
<code>string index string charIndex</code>	Returnează caracterul cu indexul <code>charIndex</code> .
<code>string last string1 string2</code>	Returnează indexul ultimei apariții a șirului <code>string1</code> în cadrul lui <code>string2</code> .
<code>string length string</code>	Returnează lungimea unui șir de caractere.
<code>string match pattern string</code>	Returnează 1 dacă modelul este echivalent cu șirul de caractere și 0 în caz contrar.
<code>string range string first last</code>	Returnează un domeniu de caractere din șirul <code>string</code> .
<code>string tolower string</code>	Convertește un șir de caractere în litere mici.
<code>string toupper string</code>	Convertește un șir de caractere în majuscule.
<code>string trim string chars</code>	Returnează șirul după eliminarea unor caractere de la începutul sau de la sfârșitul său.
<code>string trimleft string chars</code>	Returnează șirul după eliminarea unor caractere de la începutul său.
<code>string trimright string chars</code>	Returnează șirul după eliminarea unor caractere de la sfârșitul său.

Tablouri

În Tcl, un *tablou* este o variabilă cu un index de tip șir de caractere. Un tablou conține elemente; indexul fiecărui element se numește *numele elementului*. Cu alte cuvinte, puteți avea acces la orice element al tabloului prin intermediul numelui acestuia. Intern, tablourile sunt implementate utilizând o structură de date extrem de eficientă, numită *tabela de dispersie (hash)*, care permite interpretorului Tcl să găsească orice element al tabloului într-un timp aproximativ constant.

Declararea unui tablou se realizează utilizând comanda `set`. În exemplul următor, am definit un tablou numit `disk_usage`, care conține spațiul ocupat pe disc de fiecare din utilizatorii sistemului:

```
set disk_usage(root) 147178
set disk_usage(naba) 28574
set disk_usage(emily) 55
set disk_usage(ivy) 60
```

După ce ați definit un tablou, puteți avea acces la elementele acestuia prin intermediul numerelor, ca în exemplul următor:

```
set user "naba"
puts "Spatiul pe disc utilizat de $user = $disk_usage($user)K"
```

Variabile de mediu

Variabilele de mediu în Tcl se găsesc într-un tablou global predefinit numit `env`, numele elementelor fiind chiar numele variabilelor. Cu alte cuvinte, puteți determina valoarea unei anumite variabile de mediu utilizând numele variabilei ca index. Comanda următoare afișează valoarea curentă a variabilei `PATH`:

```
puts "$env(PATH)"
```

Tabloul variabilelor de mediu poate fi manipulat la fel ca și celelalte variabile. Adăugarea unui nou director la variabila `PATH` se efectuează în modul următor:

```
set env(PATH) "$env(PATH):/usr/sbin"
```

Modificările survenite în tabloul variabilelor de mediu nu afectează procesul părinte (de regulă, acesta este interpretorul care a lansat scriptul Tcl). Procesele create de către script utilizând comanda `exec` vor moșteni însă variabila modificată.

Lucrul cu fișiere în Tcl

Majoritatea exemplelor prezentate până acum în acest capitol recurg la comanda `puts` a limbajului Tcl pentru afișarea rezultatelor. În mod prestabilit, `puts` scrie la ieșirea standard; aceasta este fereastra `xterm`, dacă lucrați în mediul `X`. Puteți scrie rezultate într-un fișier furnizând identificatorul fișierului ca prim argument al comenzii `puts`. Pentru obținerea identificatorului, fișierul trebuie să fie mai întâi deschis utilizând comanda `open` din Tcl. În următorul exemplu, vom deschide un fișier, vom scrie o linie în acesta și îl vom închide:

```
set fid [open "testfile" w] ; # deschidem fișierul numit testfile pentru scriere
puts $fid "Test 1..2..3" ; # utilizăm identificatorul fișierului ca parametru
al lui puts pentru a scrie în fișier
close $fid
```

Atunci când doriți să afișați un șir de caractere la ieșirea standard, nu trebuie să specificați nici un identificator de fișier. Comanda `puts` va adăuga la sfârșitul fiecărui șir afișat un caracter linie nouă. Dacă nu doriți ca la sfârșitul unui șir să fie adăugat acest caracter, utilizați comanda `puts` cu argumentul `-nonewline`, ca în exemplul următor:

```
puts -nonewline "Comanda> " ;# Opțiunea -nonewline se utilizează la afișarea
promptului de comanda
flush stdout ; # ne asigurăm ca rezultatul este imediat afișat
```

Ați remarcat de asemenea că pentru a citi o linie de la intrarea standard, am utilizat comanda `gets`. Următorul apel al comenzii `gets` citește o linie de la intrarea standard (comanda își încheie execuția la apăsarea tastei `Enter`):

```
set line [gets stdin] ;# citește o linie de la intrarea standard
```

Cuvântul cheie `stdin` reprezintă un identificator de fișier predefinit asociat intrării standard, care, în mod prestabilit, este tastatura. Alte identificatoare predefinite sunt `stdout`, pentru ieșirea standard și `stderr`, ca dispozitiv standard de afișare a erorilor. În mod prestabilit, atât `stdout` cât și `stderr` sunt conectate la ecranul calculatorului.

În exemplul următor, prezentăm o modalitate diferită de apel a funcției `gets`, pentru citirea unei linii într-o variabilă numită `line`:

```
gets stdin line ;# citește o linie în variabila numita line
```

Pentru citirea dintr-un alt fișier, trebuie mai întâi să deschideți acel fișier pentru citire, după care transmiteți identificatorul fișierului ca argument pentru `gets`. Pentru citirea tuturor liniilor din fișierul `/etc/passwd` și afișarea lor la ieșirea standard, putem proceda ca în exemplul următor:

```
set fpass [open "/etc/passwd" r] ;# deschidem fișierul /etc/passwd pentru
citire
while { [gets $fpass line] != -1 } { ;# citim toate liniile într-un ciclu while
    puts $line ;# afișăm fiecare linie în parte
}
```

Comanda `gets` este utilă la citirea fișierelor text, deoarece prelucrează o singură linie la un moment dat. De fapt, comanda caută caracterul linie nouă ca un marcaj care indică sfârșitul unei linii de text. Atunci când trebuie să citim date în format binar, trebuie să utilizăm comanda `read` în locul lui `gets`. Pentru citirea și prelucrarea unui fișier în incrementele cu lungimea de 2048 de octeți, vom utiliza comanda `read` în următorul mod:

```
# Presupunem ca fid reprezintă identificatorul unui fișier deschis
while { ![eof $fid] } { ;# până la sfârșitul fișierului
    set buffer [read $fid 2048] ;# citește în buffer 2048 de octeți
    ;# prelucrează informația din buffer
}
```

Al doilea parametru al comenzii `read` este numărul maxim de octeți care vor fi citați. Dacă omiteți acest parametru, comanda `read` va citi întregul fișier. Puteți utiliza comanda `read` în acest mod pentru a citi și prelucra dintr-o singură parcurgere un întreg fișier text. După ce ați citit conținutul fișierului, utilizați comanda `split` pentru separarea datelor în linii de text. În continuare, prezentăm un exemplu:

```
set fid [open "etc/passwd" r] ;# deschide fișierul pentru citire
set buffer [read $fid 100000] ;# citește întregul fișier în buffer
split $buffer "\n" ;# desparte bufferul în linii
foreach line $buffer {
    puts $line ;# prelucrează linia
}
```



Dacă doriți să prelucrați mai multe fișiere (de exemplu, toate fișierele cu extensia `.tcl`), utilizați comanda `glob` pentru expandarea unui nume generic de forma `*.tcl` într-o listă. Puteți apoi utiliza comanda `open` pentru deschiderea și prelucrarea fiecărui fișier în modul următor:

```
foreach filename [glob *.tcl] { ;# utilizam comanda glob pentru a crea
# lista numelor fişierelor
  puts -nonewline $filename ;# afişam numele fişierului (doar pentru test)
  set file [open $filename r] ;# deschidem fişierul
  gets $file line ;# citim prima linie
  puts $line ;# tipărim linia (doar pentru test)
# prelucrări asupra restului fişierului (daca este cazul)
  close $file ;# închidem fişierul
}
```

Acest exemplu ilustrează modul în care putem utiliza comanda `glob` într-un script.

Execuţia comenzilor sistemului Linux

Pentru a nu reproduce inutil numărul mare de comenzi Linux, limbajul Tcl oferă un mecanism prin care puteţi rula orice comandă Linux. Cunoaşterea comenzilor sistemului Linux, le puteţi utiliza direct în scripturile Tcl.

Pentru a executa o comandă Linux într-un script Tcl, se utilizează comanda `exec`. În forma cea mai simplă, comanda `exec` preia ca argument numai comanda Linux. Spre exemplu, pentru a afişa conţinutul directorului curent, tastaţi comanda:

```
exec ls
```

Rezultatul acesteia va apărea la ieşirea standard (pe ecran), adică exact ca atunci când introduceţi comanda direct de la promptul interpretorului.

La rularea comenzilor Linux din interpretor, puteţi redirecta intrarea şi ieşirea utilizând anumite caractere speciale, cum sunt `<` (pentru redirectarea intrării), `>` (pentru redirectarea ieşirii) şi `|` (pipe). Aceste opţiuni rămân valabile şi în Tcl, deoarece comanda `exec` acceptă linii întregi de comandă, care conţin inclusiv redirectări ale intrării şi ieşirii. Pentru a memora conţinutul directorului curent în fişierul numit `dirlist`, putem utiliza comanda:

```
exec ls > dirlist
```



Secret

Comanda `exec` din Tcl nu efectuează expandarea caracterelor speciale (cum sunt asteriscurile) din numele fişierelor pe care le transmite unei comenzi Linux. Pentru a utiliza asteriscuri, trebuie să efectuaţi o operaţie în plus: prelucrarea expresiei care specifică numele fişierelor cu ajutorul comenzii `glob` pentru efectuarea corectă a expandării înainte de a transmite comanda ca argument pentru `exec`. Mai mult, întreaga comandă `exec` trebuie trimisă ca argument unei comenzi `eval`. Iată un exemplu care vă permite să afişaţi lista tuturor fişierelor cu extensia `.tcl`, utilizând comanda `exec` cu `glob` şi transmitând întregul şir de comandă comenzii `eval`:

```
eval exec ls [glob *.tcl] ;# echivalentul comenzii Linux
command "ls *.tcl"
```

O introducere în Tk

Tk este o extensie a limbajului Tcl. Tk reprezintă un set de instrumente pentru X Window System, destinat realizării interfeţelor grafice prin introducerea unor comenzi cu sintaxă Tcl. După cum puteţi intui, Tk oferă un set de comenzi Tcl în afara setului predefinit. Aceste comenzi permit crearea ferestrelor, meniurilor, butoanelor şi altor componente ale interfeţelor grafice, oferind astfel o interfaţă grafică pentru script-urile Tcl.

Componentele grafice utilizate de către Tk sunt preluate din X Window System, unde sunt cunoscute sub numele de primitive. O *primitive* (*widget*) este o componentă a unei interfeţe grafice, cum ar fi un buton, o bară de defilare, un meniu, o listă sau chiar o fereastră în mod text. Primitivele din Tk au un aspect tridimensional, similar celor din Motif.



Observaţie

Dacă ştiţi să utilizaţi primitive Motif, cunoaşteţi desigur că interfaţa Motif are la bază setul de instrumente Xt Intrinsics – utilizat la construcţia primitivelor. Spre deosebire de Motif, Tk nu se bazează pe un alt set de instrumente, utilizând doar Xlib, biblioteca standard de funcţii C pentru X Window System. Avantajul este că pentru a utiliza Tk aveţi nevoie numai de X Window System, care se distribuie gratuit.

Ca şi în cazul altor cunoştinţe noi, este cel mai bine să învăţaţi Tk cu ajutorul unor exemple, pe care le veţi descoperi în secţiunile următoare.

Exemplul „Hello, World!” în Tk

Tk reprezintă o extensie suficient de importantă a limbajului Tcl pentru a dispune de propriul interpretor, care este denumit `wish` (denumire care este prescurtarea de la *windowing shell*). Interpretorul `wish` ştie să interpreteze toate comenzile predefinite în Tcl, precum şi comenzile specifice setului de instrumente Tk. Trebuie mai întâi să porniţi sistemul X pentru a putea lansa `wish`; acesta vă va permite să utilizaţi sistemul X pentru a crea interfeţe grafice.

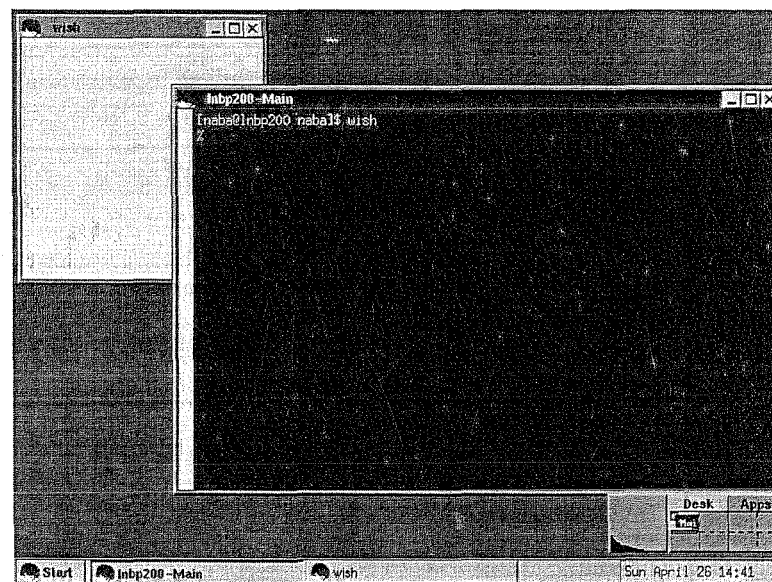
Programul `wish` se găseşte în directorul `/usr/bin`, care este inclus în mod prestabilit în variabila de mediu `PATH`. Pentru a lansa `wish`, este suficient să tastaţi următoarea comandă la promptul interpretorului într-o fereastră `xterm`:

```
wish
%
```

Programul `wish` îşi va afişa promptul (semnul procent), împreună cu o fereastră de dimensiuni mici, vizibilă în colţul din dreapta sus (în figura 8-1, această fereastră apare în stânga-sus).

Figura 8-1

Rezultatul lansării interpretorului `wish` dintr-o fereastră `xterm`.



În concluzie, interpretorul `wish` vă pune la dispoziţie un prompt interactiv, la care puteţi introduce comenzi Tk pentru a crea interfeţe grafice. Pe măsură ce `wish` va interpreta comenzile, interfaţa grafică rezultată va deveni vizibilă în fereastră.

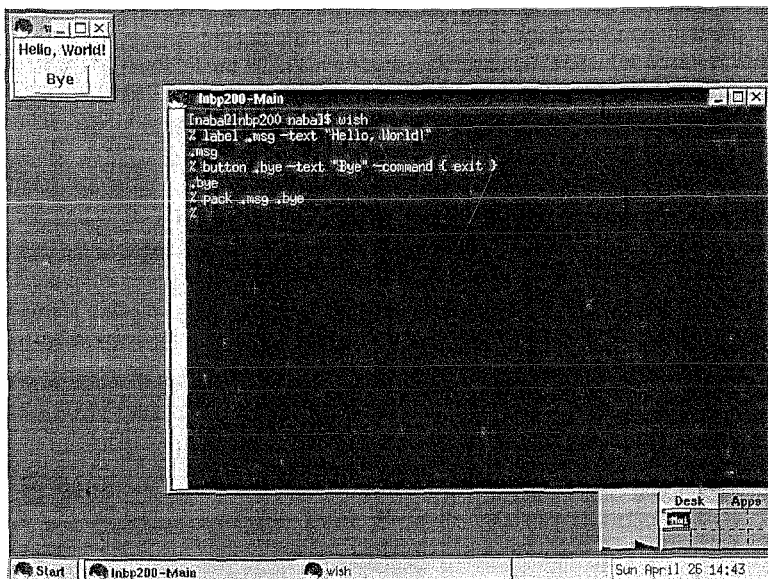
Pentru a vedea cum decurge această creare interactivă a unei interfețe grafice, introduceți următoarele comenzi la promptul wish (tastați numai ceea ce apare scris cu caractere albine):

```
% label .msg -text "Hello, World!"
.msg
% button .bye -text "Bye" -command { exit }
.bye
% pack .msg .bye
%
```

Rezultatul acestor comenzi este prezentat în figura 8-2; wish va afișa o etichetă Hello, World! sub care se găsește un buton numit Bye.

Figura 8-2

Crearea interactivă a unei etichete și a unui buton în wish.



Observați însă că eticheta și butonul apar abia după ce introduceți comanda pack. Fereastra wish se va micșora pentru a cuprinde exact eticheta și butonul.

Apăsați pe butonul Bye; programul wish se va termina. Aceasta se întâmplă din cauza argumentului command { exit } al comenzii button, care asociază acțiunii de apăsare a butonului Bye comanda Tcl exit.

Pentru a crea un program Tk în care să fie afișate eticheta Hello, World! și butonul Bye, trebuie să introduceți comenzile Tk de mai sus într-un fișier, adăugând o linie specială la începutul acestuia, pentru a vă asigura că fișierul va fi prelucrat de către interpretorul wish. Pentru aceasta, parcurgeți următoarele etape:

1. Utilizați un editor de texte pentru a scrie următoarele linii într-un fișier numit helloTk, salvând apoi fișierul:

```
#!/usr/bin/wish -f
# Un script Tk simplu
label .msg -text "Hello, World!"
button .bye -text "Bye" -command { exit }
pack .msg .bye
```



Observație

Remarcați opțiunea -f din prima linie. În versiunile de Tk anterioare lui 4.0, interpretorul wish trebuia lansat obligatoriu cu acea opțiune, pentru a accepta scriptul. Începând cu versiunea 4.0, specificarea acestei opțiuni nu mai este obligatorie.

2. Tastați următoarea comandă la promptul interpretorului pentru a acorda programului helloTk dreptul de execuție (ca rezultat al utilizării opțiunii +x a comenzii chmod):

```
chmod +x helloTk
```

3. Pentru a rula scriptul helloTk, tastați următoarea comandă la promptul interpretorului în fereastra xterm:

```
helloTk
```

Fereastra care conține eticheta Hello, World! și butonul Bye, care va apărea la rularea scriptului helloTk, este prezentată în figura 8-3. Executați clic pe butonul Bye pentru a închide fereastra, terminând execuția scriptului.

Figura 8-3

Rezultatul rulării scriptului helloTk



După cum se observă din acest exemplu, la crearea unui script Tk se parcurg aceleași etape ca și la crearea oricărui script Tcl. Singura diferență este că scripturile Tk pot afișa și componente grafice.

Principalele primitive Tk

După ce ați făcut cunoștință cu limbajul Tk, puteți trece la elaborarea unor scripturi efectiv utile. Pentru aceasta, trebuie să cunoașteți mai întâi comenzile Tk utilizate pentru crearea și configurarea primitivelor.



Observație

Termenul de *primitivă* (*widget*) are în Tk aceeași semnificație ca și în orice set de instrumente X, ca de exemplu în Motif, desemnând o componentă a unei interfețe cum ar fi un buton, o listă sau o casetă de dialog.

În exemplul din secțiunea precedentă, am utilizat o etichetă și un buton. Comanda de creare a oricărei primitive coincide cu numele primitivei respective. În consecință, comanda button creează primitive care reprezintă butoane, label creează etichete etc.

În Tk există 15 comenzi cu ajutorul cărora puteți crea primitive, comenzi prezentate în tabelul 8-4.

Tabelul 8-3 Comenzile Tk pentru crearea primitivelor

Comandă	Acțiune executată
button	Creează un buton.
canvas	Creează o suprafață de desenare în care puteți afișa text, imagini bitmap, linii, casete, poligoane și alte primitive.
checkboxbutton	Creează un buton cu două stări și îi asociază o variabilă Tcl.
entry	Creează o primitivă în care se poate introduce text în mod online.
frame	Creează un cadru capabil de a conține alte primitive.
label	Creează o primitivă reprezentând o etichetă care nu poate fi modificată.
listbox	Creează o listă în care pot defila mai multe linii de text.

(continuare)

Tabelul 8-3 Continuare

Comandă	Acțiune executată
menu	Creează un meniu.
menubutton	Creează un buton care generează un meniu asociat atunci când este acționat.
message	Creează o primitivă reprezentând un mesaj pe mai multe linii al cărui text nu poate fi modificat prin acțiunea utilizatorului.
radiobutton	Creează un buton radio, asociat cu o variabilă Tcl.
scale	Creează o primitivă de forma unei scale, care poate ajusta valoarea unei variabile.
scrollbar	Creează o bară de defilare, care poate fi asociată altei primitive.
text	Creează o primitivă de tip text, în care utilizatorul poate introduce și edita text.
toplevel	Creează o primitivă aflată pe nivelul principal (fereastra asociată acesteia este un descendent direct al ferestrei rădăcină din X Window System.)

Atunci când creați o primitivă, îi puteți specifica multe caracteristici ca argumente în linia de comandă. De exemplu, iată cum puteți crea un buton albastru etichetat cu roșu, cu șirul de caractere „test”, afișând apoi butonul:

```
button .b -text test -fg red -bg blue
pack .b
```

Comanda pack nu creează nici o primitivă. Aceasta comandă poziționează o anumită primitivă în raport cu celelalte. În tabelul 8-5, se găsesc comenzile utilizate pentru manipularea primitivelor.



Sugestie

Pentru a consulta manualul online pentru comenzile Tk din tabelele 8-4 și 8-5, tastati **man n**, apoi numele comenzii dorite. De exemplu, pentru a găsi în manualul online informații despre comanda bind, tastati **man n bind**.

Tabelul 8-4 Comenzi Tk pentru lucrul cu primitive

Comandă	Acțiune asociată
after	Execută o comandă după trecerea unei perioade de timp.
bind	Asociază o comandă Tcl cu un eveniment X, comanda fiind ulterior invocată automat la producerea evenimentului.
destroy	Distruge una sau mai multe primitive.
focus	Direcționează evenimentele de la tastatură către o anumită fereastră (acordând ferestrei respective selecția intrării).
grab	Reduce evenimentele de indicator și tastatură la o primitivă specificată și la copiii acesteia.
lower	Coboară o fereastră în <i>ordinea de stivă</i> (aceasta este ordinea în care ferestrele se suprapun una peste alta pe ecran).
option	Permite accesul la baza de resurse X.
pack	Poziționează automat primitivele în interiorul unui cadru, respectând anumite restricții.
place	Permite poziționarea manuală a unei primitive în raport cu altă primitivă.
raise	Urcă o fereastră în ordinea de stivă.

Comandă	Acțiune asociată
selection	Manipulează selecția primară X (aceasta este numele standard al unei selecții în sistemul X).
send	Trimite o comandă Tcl unei alte aplicații Tk (comandă utilizată la comunicația între procese).
tk	Afișează informații despre starea internă a interpretorului Tk.
tkerror	Tratează erorile care apar în aplicațiile Tk (interpretorul apelează această comandă la fiecare eroare pe care o detectează într-o aplicație Tk).
tkwait	Așteaptă producerea unui eveniment cum ar fi distrugerea unei ferestre sau modificarea valorii unei variabile.
update	Prelucrează toate evenimentele aflate în așteptare și actualizează imaginea de pe ecran.
wininfo	Returnează informații despre o anumită primitivă.
wm	Permite accesul la programul de gestiune a ferestrelor. (Puteți de exemplu să trimiteți anumite comenzi programului de gestiune a ferestrelor, impunând o anumită dimensiune pentru o fereastră aflată pe nivelul superior.)

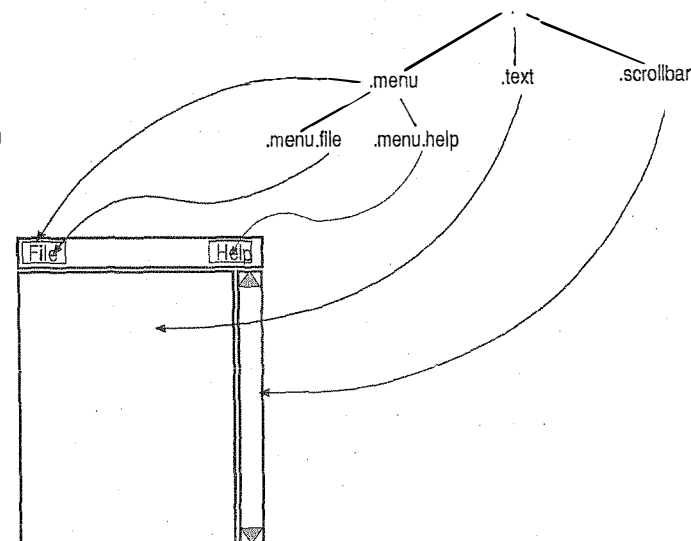
Convenția de nume a primitivelor

Din exemplul în care am creat o etichetă și un buton, puteți intui că argumentul care urmează după numele comenzii de creare este chiar numele primitivei. Dacă vă întrebați de ce toate numele încep obligatoriu cu un punct, absolut necesar la începutul numelui unei primitive, aceasta se întâmplă din cauza faptului că primitivele sunt organizate într-o ierarhie.

Să presupunem că avem o fereastră principală în care apar o bară de meniuri, o zonă în care putem introduce text și o bară de derulare. Bara de meniuri are două opțiuni, etichetate cu **File** și **Help**. În figura 8-4, puteți vedea atât aspectul pe ecran al acestei ierarhii, cât și relația dintre numele primitivelor și poziția lor în cadrul ierarhiei.

Figura 8-4

Relația dintre numele primitivelor și poziția acestora în cadrul ierarhiei



Rădăcina ierarhiei este chiar fereastra principală a aplicației. Aceasta este denumită utilizând un singur punct (.). Această fereastră este un descendent direct al ferestrei rădăcină asociate ecranului X. Toți descendenții direcți al ferestrei principale au nume care încep cu un punct. Spre exemplu, `.menu` este un descendent al ferestrei principale.

Numele celorlalte primitive depind de pozițiile acestora în cadrul ierarhiei. Butoanele din bara de meniuri au nume proprii - `.menu.file` și `.menu.help` - care indică faptul că sunt descendenții primitivei `.menu`. Numele primitivelor sunt stabilite pe măsură ce acestea sunt create, definindu-se ierarhia în care se găsesc.

Printr-o analogie, mecanismul de denumire a primitivelor este similar cu calea în care se găsește un fișier în Linux. Punctul din numele primitivelor este analogul caracterului slash (/) din calea completă a unui fișier. În documentația Tk și în manualul online, se întrebuințează denumirea de *cale Tk* pentru a indica ierarhia de primitive.



Observație

Toate numele de primitive trebuie să înceapă cu o literă mică sau cu o cifră. Numele nu pot conține puncte, acestea indicând poziția primitivelor în cadrul ierarhiei. Numele care încep cu litere mari denotă clase utilizate pentru specificarea resurselor. Semnificația termenului de resurse este aceeași ca și în X.



Referință încrucișată

Vezi capitolul 6 pentru o discuție referitoare la resurse X și la modul de specificare a acestora.

Configurarea primitivelor

În Tk, numele tuturor primitivelor sunt tratate ca nume de comenzi. Puteți efectua operații asupra unei anumite primitive utilizând numele primitivei, urmat de o listă de argumente semnificative. De exemplu, dacă avem un buton numit `.b`, putem utiliza următoarea comandă pentru a-i schimba culoarea de fond în albastru:

```
.b config -fg blue
```

Putem modifica și textul afișat de buton utilizând comanda următoare:

```
.b config -text Goodbye!
```

Perechile de argumente `-fg blue` și `-text Goodbye!` specifică atributele primitivei. Numele acestor atribute încep cu o cratimă, ca în cazul `-text`. Argumentul imediat următor este valoarea atributului respectiv.

Afișarea primitivelor

În Tk, o primitivă nu este afișată până când nu utilizați o comandă pentru poziționarea ei în fereastra principală. Trebuie să utilizați *procedura de gestiune a geometriei* – o procedură Tk al cărei rol este de a dispune una sau mai multe primitive descendent într-o primitivă părinte.

În Tk sunt disponibile două comenzi de gestiune a geometriei:

- Comanda `place`, care vă permite să poziționați o primitivă la o locație precisă din fereastra unei primitive principale (aceasta nu trebuie neapărat să fie o primitivă părinte). Primitiva principală este utilizată cu rol de referință – celelalte primitive sunt poziționate relativ la aceasta. Puteți de asemenea să precizați poziții relative, cum ar fi un deplasament pe orizontală egal cu jumătate din lățimea primitivei principale.

- Comanda `pack`. Această comandă dispune primitivele copil pe laturile primitivei principale. Pentru fiecare primitivă copil, puteți preciza latura părintelui pe care să fie poziționată aceasta,

cât și spațiul liber pe care doriți să îl lăsați în jurul său. După cum o sugerează și numele, comanda `pack` va împacheta toate primitivele cât mai strâns posibil.

Chiar dacă atât comanda `place` cât și `pack` suportă multe opțiuni diferite, utilizarea lor este destul de simplă. Vă recomandăm să lansați interpretorul `wish`, să creați câteva primitive, încercând apoi comenzile `place` și `pack` pentru a vedea ce efecte au acestea. După aceea, puteți consulta manualul online pentru a determina în mod exact opțiunile de care aveți nevoie.



Sugestie

Comenzile `pack` și `place` dispun ambele de o formă - `pack forget` și respectiv `place forget` - la care puteți recurge pentru a ascunde o primitivă. Dacă doriți, de exemplu, să ascundeți un buton numit `.btn1`, utilizați comanda `pack forget .btn1`. Pentru a determina reapariția butonului, introduceți comanda `pack .btn1` încă o dată.

Comanda pack

Comanda `pack` este cea mai răspândită comandă pentru managementul geometriei în Tk. După cum am văzut în câteva exemple simple de la începutul acestui capitol, pentru a determina apariția unui buton numit `.btn1`, utilizați forma următoare a comenzii `pack`:

```
pack .btn1
```

Puteți de asemenea să specificați numele mai multor primitive în aceeași comandă `pack`. Pentru a afișa butoanele `.btn1`, `.btn2` și `.btn3` dispuse pe o linie verticală în această ordine, introduceți următoarea comandă:

```
pack .btn1 .btn2 .btn3
```

Sintaxa comenzii `pack` este rezumată în tabelul 8-5. După cum puteți observa, comanda vă permite atât să poziționați primitivele, cât și să obțineți anumite informații despre ierarhia lor. Dacă doriți o dispunere mai complexă a primitivelor, trebuie să utilizați opțiunile de împachetare prezentate în tabelul 8-6.

Tabelul 8-5 Diferite forme ale comenzii pack

Comandă	Descriere
<code>pack widgetNames options</code>	Împachetează primitivele selectate respectând opțiunile precizate (la fel ca și <code>pack configure</code>). Lista opțiunilor disponibile se găsește în tabelul 8-6.
<code>pack configure widgetNames options</code>	Împachetează primitivele listate în conformitate cu opțiunile precizate.
<code>pack forget widgetNames</code>	Ascunde primitivele precizate.
<code>pack info widget</code>	Returnează configurația de împachetare (o listă de opțiuni și valori) primitivei specificate.
<code>pack propagate widget boolean</code>	Dacă argumentul <code>boolean</code> este 1, permite propagarea geometriei primitivei specificate; altfel, dezactivează propagarea. (Atunci când propagarea geometriei este activă, mărimea ferestrei unei primitive este determinată de dimensiunile primitivelor pe care le conține.)
<code>pack slaves widget</code>	Returnează lista primitivelor gestionate de primitiva specificată.

Tabelul 8-6 Opțiuni de împachetare a primitivelor

Opțiunea	Descriere
-after <i>widgetName</i>	Plasează primitiva împachetată imediat după primitiva precizată prin argumentul <i>widgetName</i> .
-anchor <i>anchorPos</i>	Determină poziția unei primitive gestionate. (Se aplică numai dacă primitiva container este mai mare decât cea gestionată.) Valoarea argumentului <i>anchorPos</i> poate fi center, e, n, ne, nw, s, se, sw sau w; valoarea prestabilită este center.
-before <i>widgetName</i>	Plasează primitiva împachetată imediat înaintea primitivei precizate prin argumentul <i>widgetName</i> .
-expand <i>boolean</i>	Dacă argumentul <i>boolean</i> are valoarea 1, primitiva inclusă va fi extinsă până când va ocupa tot spațiul disponibil în primitiva care o conține.
-fill <i>style</i>	Indică modul de expandare a unei primitive dacă aceasta depășește dimensiunea necesară primitivelor pe care le conține. Parametrul <i>style</i> poate avea una din valorile both, none, x sau y.
-in <i>widgetName</i>	Indică primitiva în care vor fi plasate primitivele specificate în comanda pack. Această opțiune nu este necesară, fiecare primitivă fiind în mod prestabilit împachetată în părintele său.
-ipadx <i>amount</i>	Specifică spațiul suplimentar pe orizontală în interiorul primitivei care va fi împachetată (în plus față de spațiul de care aceasta are oricum nevoie). Valoarea <i>amount</i> este un număr și se exprimă în unități de afișare pe ecran.
-ipady <i>amount</i>	Specifică spațiul suplimentar pe verticală în interiorul primitivei care va fi împachetată (în plus față de spațiul de care aceasta are nevoie). Valoarea <i>amount</i> este un număr și se exprimă în unități de afișare pe ecran.
-padx <i>amount</i>	Specifică spațiul suplimentar pe orizontală în exteriorul marginii primitivei care va fi împachetată (în plus față de spațiul de care aceasta are nevoie). Valoarea <i>amount</i> este un număr și se exprimă în unități de afișare pe ecran.
-pady <i>amount</i>	Specifică spațiul suplimentar pe verticală în exteriorul marginii primitivei care va fi împachetată (în plus față de spațiul de care aceasta are oricum nevoie). Valoarea <i>amount</i> este un număr și se exprimă în unități de afișare pe ecran.
-side <i>sideName</i>	Împachetează pe partea specificată. Valoarea argumentului <i>sideName</i> poate fi bottom, left, right sau top; valoarea prestabilită este top.



Sugestie

Dacă vă puneți întrebarea cum veți reuși să rețineți toate aceste opțiuni, sfatul nostru este să *nu* le rețineți. Vă puteți descurca pentru început utilizând doar câteva dintre ele; după un timp, le veți reține de la sine. Pentru a vă familiariza cu rolul fiecărei opțiuni, lansați interpretorul wish, creați câteva primitive și încercați să le împachetați utilizând diferite opțiuni. De acum înainte, când aveți nevoie de

sintaxa exactă a unei anumite opțiuni, consultați manualul online tastând `man n pack` sau mai simplu `man pack`.

Comanda place

Comanda `place` ne oferă o modalitate mai simplă de specificare a poziției primitivelor. Dezavantajul acesteia este că trebuie să poziționați chiar dumneavoastră toate ferestrele. Comanda este mai simplă decât `pack`, deoarece vă oferă un control direct asupra poziției primitivelor. Pe de altă parte, controlul direct al poziției primitivelor este convenabil atunci când numărul ferestrelor este mic, dar devine complicat dacă numărul primitivelor care compun interfața grafică este mai mare.

Cu ajutorul comenzii `place`, puteți poziționa o primitivă la o locație bine precizată. Pentru plasarea butonului `.btn1` la coordonatele (100, 50) în cadrul primitivei părinte, utilizați următoarea comandă:

```
place .btn1 -x 100 -y 50
```

Comanda `place` este extrem de utilă dacă trebuie să centram o anumită primitivă în raport cu părintele său. Pentru aceasta, utilizați opțiunile `-relx` și `-rely` ale comenzii `place`. Dacă `.f` este ● primitivă cadru (care conține alte primitive), îl putem afișa centrat față de fereastra părinte utilizând următoarele comenzi:

```
frame .f
button .f.b1 -text Configure
button .f.b2 -text Quit
pack .f.b1 .f.b2 -side left
place .f -relx 0.5 -rely 0.5 -anchor center
```

După cum puteți observa în acest exemplu, împachetarea butoanelor în interiorul primitivei cadru se face cu comanda `pack`. Numai cadrul `.f` este poziționat utilizând `place`. Opțiunile `-relx` și `-rely` vă permit să precizați pozițiile relative sub formă de fracții din dimensiunea primitivei părinte. Valoarea zero pentru `-relx` determină o poziționare la marginea din stânga, 1 corespunde marginii din dreapta, în timp ce 0.5 va poziționa primitiva centrată.

Ca și `pack`, și comanda `place` are mai multe forme, listate în tabelul 8-7. De asemenea, `place` poate lua mai multe opțiuni, care se regăsesc în tabelul 8-8. Puteți utiliza aceste opțiuni atât în cadrul comenzii `place`, cât și al comenzii compuse `place configure`.

Tabelul 8-7 Formele comenzii place

Comandă	Descriere
<code>place widgetName options</code>	Poziționează primitivele listate respectând opțiunile specificate (la fel ca și <code>place configure</code>). Lista opțiunilor disponibile se găsește în tabelul 8-8.
<code>place configure widgetName options</code>	Poziționează primitivele listate respectând opțiunile specificate.
<code>place forget widgetName</code>	Nu mai gestionează primitivele și le ascunde.
<code>place info widget</code>	Returnează lista de opțiuni și valorile acestora pentru primitiva precizată.
<code>place slaves widget</code>	Returnează lista primitivelor gestionate de primitiva precizată.

Tabelul 8-8 Opțiuni de plasare a primitivelor

Opțiune	Descriere
-anchor <i>anchorPos</i>	Precizează ce punct al primitivei gestionate va fi plasat în poziția specificată din fereastra părinte. Valoarea argumentului <i>anchorPos</i> poate fi center, e, n, ne, nw, s, se, sw sau w; valoarea prestabilită este nw (colțul din stânga sus).
-bordermode <i>bmode</i>	Indică modul în care sunt utilizate marginile ferestrei părinte la poziționarea unei primitive. Valoarea argumentului <i>bmode</i> poate fi ignore, inside sau outside.
-height <i>size</i>	Precizează înălțimea primitivei gestionate.
-in <i>widgetName</i>	Indică primitiva relativ la care vor fi precizate pozițiile primitivelor specificate în comanda <i>place</i> . Dacă omiteți această opțiune, primitivele vor fi plasate relativ la primitivele părinte (.f.b va fi poziționat relativ la .f).
-relheight <i>fraction</i>	Specifică înălțimea primitivei gestionate ca fracție din înălțimea primitivei părinte. Frația este o valoare în virgulă mobilă.
-relwidth <i>fraction</i>	Specifică lățimea primitivei gestionate ca fracție din lățimea primitivei părinte. Frația este o valoare în virgulă mobilă.
-relx <i>fraction</i>	Specifică poziția pe orizontală a primitivei gestionate ca fracție din lățimea primitivei părinte. Frația are o valoare în virgulă mobilă: o valoare de 0.0 corespunde marginii din stânga, în timp ce 1.0 corespunde marginii din dreapta.
-rely <i>fraction</i>	Specifică poziția pe verticală a primitivei gestionate ca fracție din înălțimea primitivei părinte. Frația are o valoare în virgulă mobilă: o valoare de 0.0 corespunde marginii de sus, în timp ce 1.0 corespunde marginii de jos.
-x <i>coord</i>	Precizează poziția pe orizontală a punctului de referință al primitivei gestionate în cadrul primitivei părinte. Valoarea <i>coord</i> se exprimă în sistemul de referință al ecranului.
-y <i>coord</i>	Precizează poziția pe verticală a punctului de referință al primitivei gestionate în cadrul primitivei părinte. Valoarea <i>coord</i> se exprimă în sistemul de referință al ecranului.
-width <i>size</i>	Precizează lățimea primitivei gestionate.

Asocierea acțiunilor cu evenimente

Atunci când scriem un program cu o interfață grafică, putem utiliza evenimente cum sunt apăsarea unor taste sau apăsarea butoanelor mouse-ului pentru inițierea unor activități în cadrul programului. Într-un script Tk, putem stabili ce activități va efectua programul prin asocierea acțiunilor cu evenimentele. În acest caz, *acțiunea* este un script Tcl care poate executa o anumită activitate. Spre exemplu, un buton etichetat cu Quit va avea asociată ca acțiune comanda Tcl *exit*, care încheie execuția scriptului Tk.

Pentru cazul butoanelor, o modalitate simplă de a forma o asociere este executând clic pe buton. Utilizați opțiunea *-command* a comenzii *button* pentru a preciza comanda Tcl care va fi executată atunci când utilizatorul apasă pe buton. Asocierea dintre comanda *exit* și butonul Quit se poate face în modul următor:

```
button .b -text Quit -command { exit }
```



Sugestie

Acoladele nu sunt necesare atunci când avem o singură comandă Tcl; utilitatea acestora apare la gruparea mai multor comenzi.

Modalitatea cea mai răspândită de a asocia o acțiune cu un eveniment este prin intermediul comenzii *bind*. Sintaxa generală a comenzii *bind* este următoarea:

```
bind nume_widget <specificatie_eventiment> Comanda_Tcl
```

Argumentul *nume_widget* este de obicei calea către un anumit widget, dar puteți asocia un eveniment unei clase de primitive, de exemplu tuturor butoanelor. Pentru precizarea parametrului *specificatie_eventiment* trebuie de regulă să consultați manualul online. (În cadrul secțiunii de față, vom prezenta mai întâi un exemplu, după care vom detalia specificațiile evenimentelor). În fine, ultimul parametru – *Comanda_Tcl* – se referă la comenzile Tcl care doriți să fie executate atunci când evenimentul are loc. Aceste comenzi Tcl pot fi orice script Tcl, de la o simplă comandă *puts* și până la un script memorat într-un fișier separat.

Pentru a ilustra printr-un exemplu mai amănunțit modul de asociere a unei acțiuni unui eveniment, să considerăm următorul scenariu. Ați remarcat că multe aplicații Microsoft Windows dispun de o bară de instrumente, care în esență este o colecție de butoane, fiecare dintre acestea fiind destinat unei activități specifice. De regulă, fiecare buton este înzestrat cu o pictogramă care îi precizează scopul. Pentru ca utilizatorii să învețe rapid semnificația butoanelor, multe aplicații Windows oferă o facilitare denumită *explicație a instrumentului*. Atunci când deplasați indicatorul de mouse deasupra unuia dintre butoane, apare o fereastră mică, în care este afișat un mesaj care explică pe scurt rolul aceluia buton.

Puteți implementa această facilitare și printr-un script Tk. Pentru aceasta, procedați astfel:

1. Creați butonul.

2. Pregătiți mesajul explicativ sub forma unei etichete (este de preferat ca aceasta să fie afișată pe un fond de culoare deschisă, de regulă galbenă).

3. Atunci când mouse-ul intră pe suprafața unui buton, are loc un eveniment de tipul Enter. Asociați evenimentul de tip Enter cu o comandă care afișează eticheta explicativă. Utilizați comanda *place* pentru a poziționa eticheta relativ la buton, astfel încât explicația să apară foarte aproape de butonul asociat. În exemplul următor, puteți vedea cum se poate asocia comanda *place* cu evenimentul Enter, utilizând o comandă *bind* corespunzătoare:

```
bind .f.b <Enter> { place .bh -in .f.b -relx 0.5 -rely 1.0 }
```

4. Atunci când butonul mouse-ului părăsește suprafața butonului, se produce evenimentul Leave. Asociați evenimentul Leave cu o comandă *place forget*, pentru a ascunde mesajul explicativ, după cum urmează:

```
bind .f.b <Leave> { place forget .bh }
```

Exemplul *toolhelp* prezentat în continuare demonstrează modul în care puteți implementa facilitarea de explicație a instrumentelor în Tk:

```
#!/usr/bin/wish
# Determina aparitia unei "ferestre explicative" atunci cand
# cursorul mouse-ului este în interiorul butonului "File"
```

```
wm geometry . 100x60
```

```
frame .f
```

```
button .f.b -text File
label .bh -text "Open file"
.bh config -bg yellow
```

```
bind .f.b <Enter> { place .bh -in .f.b -relx 0.5 -rely 1.0 }
bind .f.b <Leave> { place forget .bh }
```

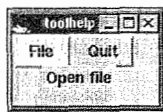
```
button .f.q -text Quit -command { exit }
```

```
pack .f.b .f.q -side left
pack .f -fill x
```

Acordați permisiunea de execuție fișierului `toolhelp` tastând `chmod +x toolhelp`. Rulați apoi scriptul tastând `toolhelp` la promptul interpretorului într-o fereastră `xterm`. În figura 8-5 este prezentată fereastra care apare după ce cursorul mouse-ului ajunge pe suprafața butonului `File`.

Figura 8-5

Un script Tk utilizat pentru afișarea mesajelor explicative



Când utilizați comanda `bind` pentru asocierea unui script Tcl cu un eveniment, trebuie să cunoașteți maniera în care puteți specifica evenimentul.

Majoritatea evenimentelor sunt produse de tastatură sau de mouse. Un număr ceva mai mic de evenimente

sunt legate de starea ferestrei asociate unei primitive. De exemplu, evenimentele `<Map>` și `<Unmap>` au loc atunci când unei primitive i se asociază o geometrie, respectiv atunci când această asociere este eliminată (ca urmare a unei comenzi `place forget` sau `pack forget`).

Evenimente de la tastatură

Există două tipuri de evenimente de la tastatură:

- `<KeyPress>` are loc la apăsarea unei taste.
- `<KeyRelease>` are loc la eliberarea unei taste.

Puteți asocia un eveniment unei anumite taste prin adăugarea `keysym`-ului (simbolului) tastei (simbolul unei taste este numele standard al tastei în X Window System) la șirul `KeyPress` și incluzând șirul rezultat între paranteze unghiulare. Spre exemplu, evenimentul asociat apăsării tastei `q` va fi specificat prin `<KeyPress-q>`. În Tk există însă o prescurtare a specificațiilor evenimentelor de la tastatură. Astfel, puteți include direct simbolul tastei între paranteze unghiulare, ca în exemplul:

```
<q>
```

Pentru majoritatea tastelor, specificația evenimentului asociat apăsării acestora este evidentă. Spre exemplu, dacă doriți încheierea scriptului atunci când utilizatorul apasă combinația `Ctrl-C` în interiorul unei primitive numite `.text`, utilizați următoarea formă a comenzii `bind`:

```
bind .text <Control-c> exit
```

Alte simboluri de taste foarte frecvent utilizate sunt și următoarele:

```
BackSpace comma Down dollar Escape exclam Left numbersign
period Return Right Tab Up
```



Sugestie

În interiorul comenzilor Tcl asociate cu un eveniment de la tastatură, utilizați cuvântul cheie `%A` pentru a vă referi la caracterul afișabil corespunzător tastei apăsate de utilizator. Pentru un caracter neafișabil, `%A` este înlocuit prin `{}` (adică o pereche de acolade între care nu se află nimic). Cuvântul cheie `%w` este înlocuit prin numele primitivei căreia îi este adresat evenimentul de la tastatură. Prin urmare, puteți insera următorul cod în interiorul unei primitive de tip `text`:

```
# Presupunem ca .text1 este un widget de tip text
bind .text <KeyPress> {
    if { "%A" != "{}" } { %W insert insert %A }
}
```

Amintiți-vă că numele unei primitive este o comandă, iar argumentele acelei comenzi depind de tipul primitivei. Pentru o primitive de tip `text`, comanda `%W insert insert %A` va insera caracterul în cadrul primitivei.

Evenimente de mouse

Utilizați numele `<ButtonPress>` și `<ButtonRelease>` pentru a denumi evenimentele de apăsare și respectiv de eliberare a oricăruia dintre butoanele mouse-ului. Pentru a preciza un eveniment specific, trebuie să adăugați la aceste denumiri numărul butonului acționat. În consecință, acțiunea de apăsare a butonului din stânga al mouse-ului (butonul cu numărul 1 în terminologia X) va fi denumită `<ButtonPress-1>`. O notație mai scurtă pentru apăsarea unui buton se obține dacă oțtem particula `Press`. Puteți deci scrie prescurtat `<Button-1>` pentru a denumi evenimentul generat de apăsarea butonului stâng al mouse-ului.



Sugestie

În cadrul comenzilor Tcl legate la un eveniment de mouse, cuvintele cheie `%x` și `%y` reprezintă coordonatele cursorului mouse-ului pe axele `x`, respectiv `y` (relativ la fereastra primitivei), în momentul producerii evenimentului de mouse. În concluzie, puteți stabili poziția la care s-a apăsă mouse-ul în interiorul unei primitive folosind un cod de forma:

```
bind .text1 <Button-1> { puts "Clic la (%x, %y) in primitiva: %W" }
```

Alte evenimente de mouse sunt `Enter` și `Leave`, care au loc atunci când deplasați cursorul mouse-ului în interiorul, respectiv în afara suprafeței unui widget. Aceste două evenimente sunt denumite `<Enter>` și `<Leave>`. Exemplul cu explicația atașată butoanelor din secțiunea „Asocierea acțiunilor cu evenimente” din acest capitol ilustrează o modalitate de utilizare a evenimentelor `<Enter>` și `<Leave>`.

Un alt eveniment legat tot de cursorul mouse-ului este `<Motion>`, care are loc la deplasarea cursorului în interiorul unei primitive.

Evenimente legate de ferestre

Pe lângă evenimentele de tastatură sau de mouse, multe alte evenimente au loc în X la manipularea unei ferestre. Serverul X generează de exemplu evenimentele `<Map>` și `<Unmap>` la afișarea, respectiv la ascunderea unei anumite primitive (utilizând comenzile `pack` sau `place`).

Atunci când utilizatorul modifică dimensiunea ferestrei, are loc un eveniment de tip `<Configure>`. Puteți deci să asociați evenimentele `<Configure>` cu o procedură de reafişare care redesenează conținutul unei primitive, pornind de la noua dimensiune a acesteia.

Mai multe informații despre X Window System

În Tk puteți capta mult mai multe evenimente X. Pentru a afla care sunt aceste evenimente și pentru alte probleme legate de X Window System, consultați una din lucrările următoare:

■ „*Graphical Applications with Tcl and Tk, Second Edition*”, de Eric Foster-Johnson, M&T Books, 1997.

■ „*X Window System Programming, Second Edition*”, de Naba Barkakati, SAMS, 1994

■ „*Volume 1: Xlib Programming Manual*”, O'Reilly & Associates, 1993

■ „*Volume 2: Xlib Reference Manual*”, O'Reilly & Associates, 1993

Evenimentele de tipul <Destroy> au loc atunci când o fereastră este pe cale de a fi distrusă. Aplicația va asocia o procedură pentru evenimentele de tip <Destroy> și va intercepta cererile de distrugere a ferestrei respective.

Rezumat

Combinatia dintre limbajul Tool Command Language (Tcl) și setul de instrumente grafice pentru X Window System al acestui limbaj, Tk, este un mediu ideal pentru dezvoltarea rapidă a aplicațiilor cu o interfață grafică. În acest capitol, mediul Tcl/Tk este prezentat cu ajutorul exemplurilor.

În urma lecturii acestui capitol, ați învățat următoarele:

- ▶ Tcl este un limbaj interpretat cu un set de comenzi care pot fi combinate respectând anumite reguli. Limbajul Tcl face parte din distribuția Red Hat Linux de pe CD-ROM-ul anexat acestei cărți. Puteți instala acest limbaj o dată cu sistemul Linux (vezi capitolul 1 pentru mai multe detalii).
- ▶ Sintaxa limbajului Tcl poate fi învățată dezvoltând scripturi Tcl în mod interactiv prin rularea interpretorului de comenzi Tcl, numit `tclsh`. Comenzile Tcl se introduc la promptul interpretorului `tclsh`.
- ▶ În Tcl există comenzi predefinite pentru majoritatea operațiilor de rutină, cum ar fi citirea și scrierea în fișiere, lucrul cu șiruri de caractere sau lansarea comenzilor Linux. În Tcl dispuneți de comenzi de control, cum ar fi `if`, `for` sau `while`, care vă permit să controlați ordinea prelucrării comenzilor de către interpretor. În fine, puteți utiliza comanda `proc` pentru a elabora noi comenzi Tcl care reprezintă combinații de comenzi existente.
- ▶ Setul de instrumente Tk este o extensie a limbajului Tcl care utilizează resursele sistemului X Window pentru a vă permite să construiți interfețe grafice. Componentele interfețelor construite în Tk au aspectul tridimensional din Motif, însă Tk nu necesită instalarea interfeței Motif sau a altor seturi de instrumente X. Tk este construit direct deasupra Xlib, interfața de programare în limbajul C sub X.
- ▶ În Tk dispuneți de comenzi pentru crearea multor primitive (componente elementare ale interfețelor grafice), cum ar fi butoane, etichete, liste și bare de derulare. Organizarea ierarhică a primitivelor se reflectă și în convenția de denumire a acestora.
- ▶ Pentru a activa o interfață grafică, trebuie să utilizați comanda `bind`, cu scopul de a asocia comenzile Tcl cu evenimente specifice de tastatură sau de mouse.
- ▶ Programele Tk pot fi realizate și testate în mod interactiv rulând `wish`, interpretorul în mod grafic care poate interpreta toate comenzile Tcl și Tk.



EXPLOATAREA ECHIPAMENTELOR HARDWARE SUB LINUX

- ▶ Capitolul 9: Calculatoare
- ▶ Capitolul 10: Plăci video și monitoare
- ▶ Capitolul 11: Unități de disc
- ▶ Capitolul 12: Unități CD-ROM și plăci de sunet
- ▶ Capitolul 13: Tastaturi și dispozitive de indicare
- ▶ Capitolul 14: Imprimante
- ▶ Capitolul 15: Modemuri
- ▶ Capitolul 16: Rețele
- ▶ Capitolul 17: PC Cards

Capitolul 9 Calculatoare

În acest capitol:

- ▶ Veți examina diferite arhitecturi ale sistemelor de calcul: ISA, EISA și MCA
- ▶ Veți folosi magistrale periferice de mare viteză VESA Local Bus (VLB) și Peripheral Component Interconnect (PCI)
- ▶ Veți folosi sistemul de fișiere `/proc` ca sursă de informații
- ▶ Veți rula Linux pe calculatoare laptop
- ▶ Veți instala Linux pe calculatoare laptop

De obicei, după ce instalați Linux folosind instrucțiunile prezentate în capitolul 1, puteți începe să-l folosiți fără a fi preocupat de arhitectura calculatorului dumneavoastră. Trebuie să luați aminte la producătorul și la modelul calculatorului și la detaliile sale interne numai dacă aveți probleme la instalare. Ca necesități minimale, trebuie să înțelegeți arhitectura hardware a calculatorului dumneavoastră – de exemplu, dacă folosește o magistrală PCI sau VLB – pentru a putea instala și folosi cu succes sistemul Linux. În cel mai rău caz, trebuie să cunoașteți detalii amănunțite despre placa de bază a sistemului dumneavoastră (placa principală de circuite a sistemului), informații cum ar fi tipul de controller IDE (Integrated Drive Electronics) folosit pe placa de bază. Alte periferice, precum hard-discul și monitorul, sunt la fel de importante; vom discuta despre aceste componente hardware în capitolele ulterioare.

Într-un fel, titlul acestui capitol este impropriu, deoarece termenul *calculator* include toate componentele acestuia, de la procesor la unitatea de disc și la monitor. Scopul acestui capitol este de a vă furniza informații cu privire la compatibilitatea sistemului Linux cu anumite tipuri de procesor și de magistrală (*magistrala* reprezintă o metodă standard de conectare pentru periferice). Deoarece un calculator laptop reprezintă un sistem mai puternic integrat decât un calculator de birou, acest capitol mai conține și informații referitoare la rularea sistemului Linux pe diverse tipuri de calculatoare laptop frecvent folosite.

Tipuri fundamentale de procesoare și de magistrale

Când treceți de la un sistem de operare simplu (precum MS-DOS) la unul mai complex (cum este Linux), detaliile exacte referitoare la componentele hardware ale calculatorului dumneavoastră devin mai importante, deoarece sistemul de operare încearcă să utilizeze toate caracteristicile acestora. Majoritatea sistemelor de operare (inclusiv Linux) folosesc programe driver de dispozitiv pentru a accesa și a folosi dispozitivele hardware. Unele componente hardware, precum procesorul sistemului și circuitele aferente, trebuie suportate de către nucleu – sistemul de operare fundamental.

Linux a fost proiectat pentru procesoare Intel 80386 sau superioare; programul folosește ceea ce este cunoscut sub numele de *mod protejat* al calculatorului. MS-DOS folosește *modul real* al procesorului, în care cantitatea de memorie adresabilă este limitată la 1 MB. În modul protejat, procesoarele 386 și superioare pot accesa o mare cantitate de memorie (teoretic, plafonul este de 4 GB).



Observație

O arhitectură obișnuită de procesor 386 limitează frecvent calculatoarele echipate cu asemenea procesoare la un plafon de 16 MB. Dacă doriți să instalați mai mult de 16 MB RAM pe un PC 386, este posibil să aveți nevoie de o modernizare ROM (Read-Only Memory) BIOS (Basic Input and Output System). Pentru aceasta, trebuie să luați legătura cu producătorul calculatorului dumneavoastră și să vă procurați noile cipuri ROM. Apoi, trebuie să eliminați vechile cipuri ROM BIOS și să le instalați pe cele noi înainte de a instala Linux. Deoarece Linux rulează foarte bine pe un 386 echipat cu suficientă memorie, modernizarea sistemului BIOS poate fi un exercițiu care merită efortul pentru a utiliza calculatorul într-un mod adecvat.

Pentru a rula Linux, aveți nevoie de un sistem echipat cel puțin cu un procesor 386. Linux rulează pe toate variantele de procesoare 80386, 80486 și Pentium. Astfel, puteți rula Linux pe procesoarele 386SX, 386DX, 386SL, 486SX, 486DX, 486DX2, 486DX4, Pentium și Pentium II de la Intel. (Această listă nu include toate variantele de procesoare 386 și 486, dar ați intuit ideea...) În mod colectiv, aceste procesoare sunt desemnate sub numele de familia de procesoare 80x86. De asemenea, Linux rulează pe procesoarele compatibile 80x86 produse de terți, precum Cyrix sau AMD.

Familia de procesoare 80x86 folosește un *procesor în virgulă mobilă* (cunoscut și sub numele de *coprocesor matematic*), pentru a accelera viteza de efectuare a calculelor în virgulă mobilă. În multe dintre procesoarele 486 și Pentium, procesorul în virgulă mobilă este o componentă a procesorului de bază. Linux folosește procesorul hardware în virgulă mobilă, dacă acesta este disponibil. În caz contrar, Linux emulează procesorul în virgulă mobilă folosind programe care fac parte din nucleul Linux.

Linux pe alte procesoare

Deși versiunea cea mai frecvent distribuită de Linux este destinată procesoarelor Intel 80x86 și Pentium, Linux este de asemenea disponibil pentru sisteme bazate pe alte procesoare, precum familia Motorola 68000; procesorul Alpha AXP al firmei Digital Equipment Corporation (DEC); procesorul Sun SPARC; procesorul Hewlett-Packard HP PA-RISC; procesorul Power PC și procesorul MIPS R4x00. Mai multe informații despre aceste versiuni de Linux sunt disponibile pe Internet. Începeți prin a vă orienta browserul spre unul sau mai multe dintre următoarele situri Web:

Linux pentru sisteme echipate cu procesoare Motorola 68000:

<http://www.clark.net/pub/lawrenc/lunix/>

Linux pentru sisteme echipate cu procesoare DEC Alpha:

<http://www.azstarnet.com/~axplinux/>
<ftp://ftp.redhat.com/pub/redhat/redhat-5.1/alpha/>

Linux pentru sisteme echipate cu procesoare Sun SPARC

http://www.geog.ubc.ca/s_linux.html
<ftp://ftp.redhat.com/pub/redhat/redhat-4.2/sparc/>

Linux pentru sisteme echipate cu procesoare HP PA-RISC:

<http://www.osf.org/mall/os/pa-mklinux/index.html>

Linux pentru sisteme echipate cu procesoare MIPS:

<http://lena.fnet.fr/>

Linux pentru sisteme echipate cu procesoare PowerPC:

<http://www.linuxppc.org/>
<http://www.mklinux.apple.com/>

Tipuri de magistrală

Termenul de *magistrală* se referă la ansamblul de fire de colectare care transportă semnale între procesorul calculatorului și un dispozitiv periferic, de orice natură ar fi acesta. După procesor, magistrala este o caracteristică hardware de o importanță esențială pentru calculator. Fiecare tip de magistrală specifică semnificația diferitelor semnale și protocolul (ordinea și durata semnalelor) pentru transferul datelor prin intermediul magistralei.

Fiecare magistrală este caracterizată de o viteză maximă de transfer al datelor, care depinde de numărul de biți de date transportabili la un moment dat, și de *frecvența de ceas*, în speță numărul de comutări între cele două stări ale semnalului (0 și 1) posibile într-o secundă. Similar cu frecvența internă de ceas a procesorului, frecvența de ceas a magistralei este exprimată în megaherți (MHz). 1 MHz înseamnă un milion de ori pe secundă.

Fiecare tip de magistrală specifică dimensiunile fizice ale plăcii controller, numărul și rolul conectorilor, precum și slotul de pe placa de bază unde se inserează placa. De exemplu, hard-discul sistemului dumneavoastră este conectat la placa de bază prin intermediul unei plăci disk-controller care este compatibilă cu tipul de magistrală al plăcii de bază respective.

În continuare sunt prezentate mai multe tipuri comune de magistrală:

■ **Magistrala ISA (Industry Standard Architecture)** – arhitectură industrială standard) a fost cea mai frecvent folosită, până la apariția magistralei PCI. Această magistrală a fost folosită în calculatorul original IBM PC-AT. Magistrala ISA poate transfera 16 biți de date simultan. Frecvența sa de ceas este 8 MHz, lentă în comparație cu procesoarele actuale, care dispun de

frecvențe de ceas din domeniul 200-300 MHz. De regulă, magistrala ISA poate realiza o viteză de transfer a datelor de 5 MB/s (adică 5 megabaiți pe secundă).

■ **Magistrala Video Electronics Standard Association (VESA) Local Bus**, sau VLB, a fost proiectată pentru un transfer de date cu performanțe ridicate între procesor și placa video. Frecvența normală de transfer pentru VLB este de 30 MB/s. Primele sisteme VLB dispun deseori de o implementare non-standard a magistralei locale VESA, fapt care poate cauza probleme la utilizarea sistemului Linux.

■ **Magistrala Micro Channel Architecture** (arhitectură micro-canal – MCA) este magistrala patentată de IBM, care a apărut pentru prima dată la calculatoarele PS/2. IBM a conceput această magistrală ca pe o magistrală de mare viteză, dar natura sa restrânsă a împiedicat utilizarea acesteia pe scară largă în PC-uri.

■ **Magistrala Extended Industry Standard Architecture (EISA)** – arhitectură industrială standard extinsă) a venit ca o alternativă la magistrala MCA, cu performanțe comparabile cu ale acesteia. Magistrala EISA nu este folosită frecvent, deoarece plăcile de periferic sunt mult mai costisitoare decât plăcile similare corespunzătoare magistralei ISA. Performanțele magistralei EISA sunt comparabile cu acelea ale magistralei VLB, transferând date la frecvențe de 30 MB/s.

■ **Magistrala Peripheral Component Interconnect (PCI)** (interconectarea componentelor periferice) este cea mai recentă magistrală de înaltă performanță; operează la o frecvență de ceas de 33 MHz și poate transfera maximum 64 de biți de date simultan. Când o magistrală PCI este folosită pentru a transfera 32 de biți simultan, frecvența de ceas de 33 MHz arată că magistrala poate transfera date la o viteză de $33 \times 4 = 132$ MB pe secundă (nu uitați că 32 biți = 4 baiți). PCI este standardul „care vine”. Generația actuală de sisteme PC Pentium folosesc magistrale PCI, dar oferă și sloturi de magistrală ISA, deci puteți continua să folosiți plăci ISA. De regulă, magistrala ISA realizează o frecvență de transfer a datelor de 60 MB/s.

Linux include suport pentru magistralele ISA, VLB, EISA și PCI; versiunea Linux de pe CD-ROM-ul anexat cărții poate fi instalată și rulată pe sisteme care dispun de oricare dintre aceste magistrale. Deoarece o magistrală are scopul de interconectare a perifericelor la sistem, suportul pentru magistrală implică posibilitatea de utilizare a perifericelor care folosesc plăci controller de tipul magistralei respective.

Desigur, nu toate componentele hardware periferice (cum sunt placa grafică și controllerul de disc) pentru toate aceste tipuri de magistrale funcționează sub Linux. Fiecare periferic necesită un driver, iar driverul este posibil să nu existe pentru unele plăci grafice sau controller de disc mai recente. Mai multe informații despre anumite dispozitive se află în capitolele 10-17.



Atenție!

Nucleul Linux existent pe distribuția Linux Red Hat inclusă pe CD-ROM-ul anexat cărții nu include suport pentru sistemele IBM PS/2 care folosesc magistrala Micro Channel Architecture (MCA). Principala problemă care apare la suportul unei magistrale de genul MCA este aceea că este necesară dezvoltarea unui software driver necesar pentru accesarea perifericelor precum unitățile de disc și plăcile video. Unii dezvoltatori Linux lucrează la suportul pentru magistrala MCA. Mai multe detalii despre Linux pe sisteme MCA se pot găsi la adresa <http://glycerine.cetmm.uni.edu/mca/>.

Suportul pentru magistrale PCI în Linux

Magistrala PCI devine rapid opțiunea preferată a tuturor calculatoarelor Pentium de producție recentă. Unul dintre elementele de atracție îl reprezintă randamentul ridicat al magistralei. De asemenea, magistrala este independentă de procesor; o placă de interfață PCI va funcționa pe un Pentium PC, dar și pe un sistem Alpha DEC, cu condiția ca ambele

sisteme să folosească o magistrală PCI. De asemenea, structura magistralei PCI permite construirea de componente PCI inserate direct pe placa de bază, reducând astfel la minimum costul *logicii de legătură* (componente electronice suplimentare, utilizate pentru a pune totul în funcțiune) necesare în alte magistrale.



Observație

Printre variantele curente de plăci de bază, noul standard pare să fie placa bazată pe cipsetul Intel Triton. Acest cipset include suport pentru un tip special de memorie, denumit EDO (Extended Data Out) DRAM (Dynamic Random Access Memory), care include un cache în cipul de memorie. Majoritatea calculatoarelor curente Pentium PC folosesc plăci de bază Intel Triton cu EDO RAM și un cache de 512KB. De obicei, o placă PCI dispune de un cipset video PCI incorporat și de o interfață IDE (Integrated Drive Electronics) pentru conectarea hard-discurilor IDE la sistem. Interfața PCI IDE va fi descrisă ulterior în capitolul 11.

XFree86 include suport pentru multe plăci video PCI – cum ar fi ATI Graphics Pro Turbo, Diamond Viper PCI și Number Nine GXE Pro PCI – precum și cipseturi video integrate, ca S3 Trio32/Trio64.



CD

Versiunea Linux de pe CD-ROM-ul anexat cărții include codul PCI Probe, care colectează informații despre dispozitivele PCI din sistemul dumneavoastră la încărcarea sistemului de operare Linux. Veți avea acces la această informație prin emiterea unei comenzi cat `/proc/pci`. (Sistemul de fișiere `/proc` va fi explicat în secțiunea „Informații din sistemul de fișiere `/proc`”).

Unele probleme specifice

Când aveți probleme cu instalarea și configurarea sistemului Linux pe PC-ul dumneavoastră, majoritatea acestora sunt legate de diferite periferice, în speță controllere de disc, plăci video, interfața cu unitatea CD-ROM etc. Cu toate acestea, există probleme cu referință directă la procesor sau la arhitectura hardware elementară a calculatorului. În această secțiune vom discuta despre câteva asemenea probleme și vom propune soluții.

Reducerea vitezei sistemului după adăugarea de memorie

Această problemă survine după ce adăugați un plus de memorie la sistemul dumneavoastră (de la 16 MB la 32 MB, de exemplu) și vă așteptați la o îmbunătățire semnificativă a performanțelor; dimpotrivă, viteza sistemului se reduce drastic. Cauza acestui comportament bizar este aceea că, atunci când adăugați memorie, trebuie să asigurați actualizarea memoriei cache.

Memoria cache este o memorie rapidă (și costisitoare) care menține viteza maximă de lucru a procesorului. (*Viteza memoriei* se referă la viteza cu care procesorul poate scrie, respectiv citi date din memoria respectivă.) De obicei, un sistem poate avea 256 K sau 512 K de memorie cache.

Memoria obișnuită (aflată în sistem sub forma de puteri ale lui 2 – 16, 32, 64 MB etc.) este prea lentă pentru a putea ține pasul cu procesoarele rapide actuale. Memoria cache se comportă ca un mediu de stocare intermediar între memoria normală și procesor.



Secret

Când adăugați memorie, trebuie să vă asigurați că memoria cache lucrează cu noua cantitate de memorie. Soluția exactă depinde de placa de bază. Uneori, soluția se reduce la rularea programului de configurare al calculatorului și activarea unei opțiuni de cache a noii zone de memorie, opțiune care era anterior dezactivată.

Pe unele sisteme, memoria trebuie instalată în anumite socluri fizice pentru a garanta cache-ul pentru întreaga cantitate de memorie. O altă soluție mai poate fi setarea *jumpereleor* – conectori între pinii de pe placa de bază – pentru a activa caching-ul.

Unele plăci de bază pot crea un cache pentru un anumit volum de memorie, în funcție de cantitatea de memorie cache. De regulă, o placă de bază cu 256 K de memorie cache este capabilă de a satisface necesitățile de caching ale sistemului. Chiar și cu un cache de 256 K poate fi necesară setarea unor jumpere sau activarea unor opțiuni, pentru a permite caching pentru toată memoria. Consultați documentația aferentă sistemului pentru a găsi indicii; informația de care aveți nevoie se poate găsi în documentația aferentă plăcii de bază.



Secret

Dacă adăugați memorie dincolo de 64 MB, sistemul își reduce viteza deoarece Linux nu poate determina cantitatea exactă de memorie de care dispune calculatorul. La inițializarea sistemului, Linux folosește apelurile BIOS pentru a determina volumul de memorie instalată. Din păcate, apelurile BIOS pot raporta maximum 64 MB memorie instalată. Dacă dispuneți de peste 64 MB de memorie, trebuie să specificați această cantitate prin intermediul unei opțiuni de boot LILO. De exemplu, dacă sistemul dumneavoastră dispune de 96 MB de memorie, adăugați următoarea opțiune la promptul de boot:

Mem=96M



Secret

Când specificați cantitatea de memorie la executarea operației de boot, nu uitați că unele PC-uri pot folosi ultimele 384 KB de memorie pentru a executa cache pe BIOS (în esență, codul BIOS este copiat în această regiune). Pentru a fi sigur, în cazul unui PC cu 96 MB de memorie, puteți folosi opțiunea `mem=95M`.

Sistemul Cyrix 6x86 are unele probleme

Familia de procesoare Cyrix 6x86 este o clonă a familiei de procesoare Intel x86, cu performanțe comparabile cu ale procesoarelor Intel Pentium. Ca atare, Linux funcționează pe sisteme Cyrix 6x86. Există însă și probleme legate de sistemul 6x86.



Secret

O scurtă secvență de instrucțiuni poate determina procesorul 6x86 să înceteze a mai răspunde la întreruperi. În esență, sistemul „îngheață”. Puteți afla mai multe despre această problemă la adresa <http://www.tux.org/~balsa/linux/cyrix/p11.html>.

Soluția este de a activa bitul `NO_LOCK` din registrul de configurare CCR1. Un program denumit `set6x86` este disponibil pentru a corecta problema. Acest program mai poate activa o caracteristică a procesorului 6x86, care întrerupe consumul de energie electrică atunci când procesorul este inactiv.

Mai există unele probleme minore, cum ar fi calculul greșit al valorii `BogoMips` pe procesoarele Cyrix 6x86. Pentru mai multe informații privind Linux și procesoarele Cyrix 6x86, vizitați situl Web <http://www.tux.org/~balsa/linux/cyrix/>. În acest sit, mai puteți descoperi toate corecțiile necesare pentru procesoarele Cyrix 6x86.

Informații din sistemul de fișiere /proc

Puteți descoperi foarte multe detalii despre calculatorul dumneavoastră prin consultarea conținutului unui sistem de fișiere special, cunoscut sub numele de sistem de fișiere `/proc`. Bineînțeles, pentru a putea folosi acest sistem de fișiere, trebuie să aveți instalat sistemul Linux pe calculatorul dumneavoastră. Cunoașterea unor date despre sistemul de fișiere `/proc` este utilă, deoarece vă poate ajuta să determinați cu exactitate informația despre calculatorul dumneavoastră (sau, cel puțin, ceea ce crede Linux cu privire la sistemul dumneavoastră) în

cazul în care adăugați un dispozitiv nou sau regenerați nucleul și încercați să decideți care sunt tipurile de caracteristici ale nucleului care trebuie activate.

Sistemul de fișiere `/proc` nu este un director real pe disc, ci o colecție de structuri de date în memorie, gestionate de nucleul Linux, care apare utilizatorului sub formă de directoare și fișiere. Scopul sistemului `/proc` (cunoscut și sub numele de *sistem de fișiere proces*) este de a permite utilizatorului să acceseze informații referitoare la nucleul Linux și la procesele care rulează în mod curent pe sistemul dumneavoastră.

Puteți accesa sistemul de fișiere `/proc` așa cum accesați orice alt director, dar trebuie să cunoașteți semnificația diferitelor fișiere pentru a putea interpreta informația. De regulă, puteți folosi comanda `cat` sau `more` pentru a vizualiza conținutul unui fișier din `/proc`; conținutul fișierului furnizează informații referitoare la anumite aspecte din sistem.

Ca în cazul oricărui alt director, puteți începe prin a examina un listing detaliat al conținutului acestuia. Pentru aceasta, tastați `ls -l /proc`. În continuare este prezentat rezultatul acestei comenzi pe sistemul meu:

```
ls -l /proc
total 0
dr-xr-xr-x 3 root root 0 May 3 06:13 1
dr-xr-xr-x 3 root root 0 May 3 06:13 1623
dr-xr-xr-x 3 pekkle16 users 0 May 3 06:13 1624
dr-xr-xr-x 3 pekkle16 users 0 May 3 06:13 1625
dr-xr-xr-x 3 root root 0 May 3 06:13 1636
dr-xr-xr-x 3 root root 0 May 3 06:13 1637
dr-xr-xr-x 3 root root 0 May 3 06:13 1670
dr-xr-xr-x 3 root root 0 May 3 06:13 1702
dr-xr-xr-x 3 naba users 0 May 3 06:13 1703
dr-xr-xr-x 3 naba users 0 May 3 06:13 1704
dr-xr-xr-x 3 root root 0 May 3 06:13 1719
dr-xr-xr-x 3 root root 0 May 3 06:13 1720
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1783
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1784
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1785
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1786
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1787
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1788
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1789
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1790
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1791
dr-xr-xr-x 3 nobody nobody 0 May 3 06:13 1792
dr-xr-xr-x 3 root root 0 May 3 06:13 187
dr-xr-xr-x 3 root root 0 May 3 06:13 196
dr-xr-xr-x 3 root root 0 May 3 06:13 2
dr-xr-xr-x 3 root root 0 May 3 06:13 2000
dr-xr-xr-x 3 daemon root 0 May 3 06:13 207
dr-xr-xr-x 3 root root 0 May 3 06:13 218
dr-xr-xr-x 3 root root 0 May 3 06:13 22
dr-xr-xr-x 3 root root 0 May 3 06:13 230
dr-xr-xr-x 3 root root 0 May 3 06:13 241
dr-xr-xr-x 3 root root 0 May 3 06:13 252
dr-xr-xr-x 3 root root 0 May 3 06:13 267
```



```

dr-xr-xr-x 3 root    root    0 May 3 06:13 279
dr-xr-xr-x 3 root    nobody  0 May 3 06:13 290
dr-xr-xr-x 3 root    root     0 May 3 06:13 3
dr-xr-xr-x 3 postgres postgres 0 May 3 06:13 316
dr-xr-xr-x 3 root    root     0 May 3 06:13 332
dr-xr-xr-x 3 root    root     0 May 3 06:13 341
dr-xr-xr-x 3 root    root     0 May 3 06:13 382
dr-xr-xr-x 3 root    root     0 May 3 06:13 383
dr-xr-xr-x 3 root    root     0 May 3 06:13 384
dr-xr-xr-x 3 root    root     0 May 3 06:13 385
dr-xr-xr-x 3 root    root     0 May 3 06:13 386
dr-xr-xr-x 3 root    root     0 May 3 06:13 387
dr-xr-xr-x 3 root    root     0 May 3 06:13 389
-r--r--r-- 1 root    root     0 May 3 06:13 cmdline
-r--r--r-- 1 root    root     0 May 3 06:13 cpuinfo
-r--r--r-- 1 root    root     0 May 3 06:13 devices
-r--r--r-- 1 root    root     0 May 3 06:13 dma
-r--r--r-- 1 root    root     0 May 3 06:13 filesystems
-r--r--r-- 1 root    root     0 May 3 06:13 interrupts
-r--r--r-- 1 root    root     0 May 3 06:13 ioports
-r----- 1 root    root    1678312 May 3 06:13 kcore
-r----- 1 root    root     0 Apr 28 03:39 kmsg
-r--r--r-- 1 root    root     0 May 3 06:13 ksyms
-r--r--r-- 1 root    root     0 May 3 02:39 loadavg
-r--r--r-- 1 root    root     0 May 3 06:13 locks
-r--r--r-- 1 root    root     0 May 3 06:13 mdstat
-r--r--r-- 1 root    root     0 May 3 06:13 meminfo
-r--r--r-- 1 root    root     0 May 3 06:13 misc
-r--r--r-- 1 root    root     0 May 3 06:13 modules
-r--r--r-- 1 root    root     0 May 3 06:13 mounts
dr-xr-xr-x 2 root    root     0 May 3 03:49 net
-r--r--r-- 1 root    root     0 May 3 04:23 pci
-r--r--r-- 1 root    root     0 May 3 06:13 rtc
dr-xr-xr-x 3 root    root     0 May 3 06:13 scsi
lrwxrwxrwx 1 root    root    64 May 3 06:13 self -> 2000
-r--r--r-- 1 root    root     0 May 3 06:13 stat
dr-xr-xr-x 5 root    root     0 May 3 06:13 sys
-r--r--r-- 1 root    root     0 May 3 06:13 uptime
-r--r--r-- 1 root    root     0 May 3 06:13 version

```

Primul set de directoare (indicat prin litera d de la începutul liniei) reprezintă procesul care rulează în momentul respectiv pe sistemul dumneavoastră. Fiecare director care corespunde unui proces are drept nume identificatorul de proces (adică un număr).

Observați fișierul de mari dimensiuni /proc/kcore; acest fișier reprezintă întreaga memorie fizică a sistemului dumneavoastră. Așa cum se va arăta în următoarele secțiuni, fișierele din sistemul /proc vă pot ajuta să aflați o mulțime de informații despre sistemul dumneavoastră.

Fișierul /proc/cpuinfo

Mai multe fișiere din /proc conțin informații interesante referitoare la caracteristicile hardware ale sistemului dumneavoastră. De exemplu, fișierul /proc/cpuinfo prezintă caracteristicile

esențiale ale sistemului dumneavoastră, cum ar fi tipul procesorului și informații despre procesorul în virgulă mobilă.

Pe unul dintre sistemele mele, când doresc să examinez fișierul /proc/cpuinfo, tastez comanda `ls /proc/cpuinfo` și voi obține următoarele date de ieșire:

```

cat /proc/cpuinfo
processor       : 0
cpu           : 586
model          : Pentium 75+
vendor_id      : GenuineIntel
stepping       : 5
fdiv_bug       : no
hlt_bug        : no
fpu            : yes
fpu_exception  : yes
cpuid          : yes
wp             : yes
flags          : fpu vme de pse tsc msr mce cx8
bogomips       : 30.00

```

Acest listing de ieșire a fost generat de un sistem Dell Dimension P75 (Pentium la 75 MHz), produs de Dell Computer Corporation. Listingul prezintă multe caracteristici interesante ale procesorului. Observați linia care începe cu `fdiv_bug`. Vă mai amintiți de infamul defect la împărțirea cu virgulă mobilă de la procesoarele Pentium? Defectul este o instrucțiune numită *fdiv* (de la *floating point division* – împărțire cu virgulă mobilă). Astfel, linia `fdiv_bug` indică dacă acest Pentium are sau nu defectul amintit (din fericire, procesorul sistemului meu e bine sănătos).

Convenția de nume pentru procesoare

Termenul *CPU* reprezintă abrevierea de la *central processing unit* – unitate centrală de procesare. În epoca calculatorului mainframe și a minicalculelor, CPU avea aspectul unei unități (ca un dulap). Pentru un PC, totuși, CPU reprezintă microprocesorul în speță cipul 486 sau Pentium care este „inima” calculatorului. Astfel, fișierul care conține informații despre procesor (CPU) se numește `cpuinfo` – un nume derivat din concatenarea termenului `cpu` cu `info` (prescurtarea la informație). Am explicat acest proces de gândire, deoarece acesta este modul în care programatorii creează nume scurte, misterioase. După ce vă veți familiariza cu unele dintre aceste nume, veți începe să le deduceți mai rapid semnificația.



Referință încrucișată

Ultima linie din fișierul /proc/cpuinfo prezintă valoarea `BogoMips` (pentru mai multe informații despre `BogoMips`, citiți textul suplimentar „Ce este `BogoMips`?” din capitolul 1) pentru procesor, așa cum a fost calculată de nucleul Linux la încărcare. `BogoMips` este o entitate folosită intern de Linux pentru programarea ciclurilor de întârziere.

Fișierul /proc/pci

Dacă dispuneți de un sistem PCI (caracteristic majorității configurațiilor Pentium), puteți examina fișierul /proc/pci pentru a afla informații despre dispozitivele PCI amplasate pe sistemul dumneavoastră. Ca de obicei, tasteți `cat /proc/pci` pentru a vedea informațiile. În continuare, iată rezultatul comenzii anterioare aplicate unui sistem Dell Pentium cu o placă de bază PCI:

```

PCI devices found:
Bus 0, device 13, function 0:
IDE interface: CMD 640(buggy) (rev 0).

```

```

Medium devsel. IRQ 14.
Bus 0, device 10, function 0:
VGA compatible controller: S3 Inc. Trio32/Trio64 (rev 0).
Medium devsel. IRQ 255.
Non-prefetchable 32 bit memory at 0xfe000000.
Bus 0, device 1, function 0:
ISA bridge: VLSI 82C593-FC1 (rev 1).
Medium devsel. Fast back-to-back capable. Master Capable. No bursts.
Bus 0, device 0, function 0:
Host bridge: VLSI 82C592-FC1 (rev 1).
Medium devsel. Fast back-to-back capable. Master Capable. Latency=16.

```

Așa cum se vede din listingul anterior, sistemul dispune de patru dispozitive PCI. Din cele patru dispozitive, primele două sunt de interes, iar celelalte două sunt legate de interfețe între CPU și magistrala ISA.



Observație

Primul dispozitiv PCI reprezintă interfața cu Integrated Drive Electronics (IDE) care sunt incluse în fiecare unitate de disc IDE. (Numele *Integrated Drive Electronics* semnifică faptul că acele componente electronice necesare pentru controlul unei unități sunt incluse în unitatea respectivă.)

Cel de-al doilea dispozitiv este controllerul de afișaj video. Acesta este o placă video (sau placă grafică) inclusă în placa de bază sub forma unui dispozitiv PCI. În esență, placa de bază conține cipurile necesare pentru controlul monitorului.

Pentru fiecare dispozitiv, fișierul `/proc/pci` mai conține numele producătorului și modelul dispozitivului. De exemplu, controllerul video este un chipset S3 Trio32/Trio64. Puteți folosi informația despre cipsetul video atunci când configurați XFree86 pentru a rula pe sistemul dumneavoastră.



Secret

Așa cum se vede în fișierul `/proc/pci`, interfața IDE a acestui sistem Pentium este CMD 640A, cunoscută ca având probleme (motiv pentru care în fișier apare specificația *buggy*). Discul sistemului și unitatea CD-ROM sunt conectate la interfața CMD640A. Deși sistemul meu nu a avut probleme cu discul sau cu unitatea CD-ROM, unii utilizatori au semnalat probleme cu controllerele IDE CMD 640 pentru magistrala PCI. În particular, se pare că interfața CMD 640 nu funcționează normal atunci când la a doua interfață sunt conectate alte unități. Versiunea Linux de pe CD-ROM-ul asociat acestei cărți dispune de corecțiile adecvate pentru remedierea neajunsurilor interfeței CMD 640A.

Alte informații incluse în sistemul de fișiere `/proc`

Toate echipamentele periferice (precum hard-discul, placa de sunet, unitatea CD-ROM, modemul și imprimanta) din sistemul dumneavoastră necesită acces la unele resurse sistem. De obicei, fiecare echipament necesită următoarele resurse:

- **numărul IRQ** (Interrupt Request – cerere de întrerupere), pe care dispozitivul îl folosește pentru a atrage atenția procesorului. IRQ se poate asimila ca fiind legătura directă între dispozitiv și procesor. Există un număr limitat de linii IRQ, deci există șanse de conflict dacă sistemul dumneavoastră conține prea multe echipamente periferice.
- **adresa Input/Output Port** (port de intrare/ieșire), folosită de procesor pentru a trimite și pentru a expedia date spre/dinspre dispozitivul periferic. Un dispozitiv necesită un domeniu unic de adrese I/O pentru a funcționa corect.

■ **canalul DMA** (Direct Memory Access – acces direct la memorie), pe care echipamentul îl folosește pentru accesarea directă a memoriei sistemului, fără a mai trece prin procesor.

Unele periferice nu necesită canale DMA, dar majoritatea lor au nevoie de un număr IRQ și de domeniul necesar de adrese I/O.

Chiar și Linux are nevoie de aceste resurse (IRQ, adrese I/O și canal DMA) pentru fiecare dispozitiv, deoarece arhitectura hardware este cea care dictează necesitatea acestor resurse. De regulă, dacă dispuneți de aceste informații disponibile pentru toate componentele hardware din sistem, puteți evita problemele care survin datorită atribuirilor conflictuale de IRQ, adrese I/O sau canale DMA.

Când re-generați nucleul (așa cum se vede în capitolul 2) pentru a adăuga suportul pentru un anumit echipament, cum ar fi o anumită placă de sunet, trebuie să furnizați informațiile necesare legate de IRQ, DMA și adresele I/O pentru a configura Linux în mod corespunzător. În cazul anumitor dispozitive, Linux poate efectua proba pentru a determina informațiile necesare.

În afară de fișierele `/proc/cpuinfo` și `/proc/pci`, mai există trei fișiere din sistemul `/proc` care conțin informații referitoare la IRQ, DMA și adresele I/O folosite în sistemul dumneavoastră.

Aceste fișiere sunt:

- `/proc/interrupts`, care conține informații despre IRQ-urile folosite
- `/proc/dma`, care prezintă lista canalelor DMA folosite
- `/proc/ioports`, care conține o listă cu domeniile de adrese I/O folosite

De exemplu, iată care este aspectul fișierului `/proc/interrupts` pentru sistemul meu:

```

0:      773590    timer
1:         10    keyboard
2:         0    cascade
4:    337181 + serial
8:         1 + rtc
10:     5447    3c509
11:     2062    aha1542
12:         0    PS/2 Mouse
13:         1    math error
14:     77969 + ide0
15:         0 + ide1

```

Primul număr de pe fiecare linie este IRQ-ul (numărul cererii de întrerupere), număr urmat de două puncte, după care apare numărul total de întreruperi (pentru acest IRQ) care au survenit până la momentul respectiv. Textul de la sfârșitul liniei indică perifericul care folosește acel IRQ. Dispozitivul 3c509 se referă la placa Ethernet 3Com509, iar aha1542 este placa controller SCSI (Small Computer System Interface) Adaptec AHA1542.

În continuare este prezentat fișierul `/proc/dma` pentru sistemul meu:

```

4: cascade
5: aha1542

```

Așa cum se vede din listing, sunt în uz numai două DMA-uri. Multe echipamente nu folosesc accesul direct la memorie.

Fișierul `/proc/ioports` conține mai multe intrări, deoarece fiecare dispozitiv necesită adrese de port I/O. Listingul următor prezintă fișierul `/proc/ioports` pentru sistemul meu (comentariile autorului sunt scrise cu caractere cursive):

0000-001f :	dma1	controller DMA (Direct Memory Access) 1
0020-003f :	pic1	Programmable Interrupt Controller (PIC) 1
0040-005f :	timer	
0060-006f :	keyboard	Tastatura
0070-007f :	rtc	Ceas in timp real
0080-009f :	dma page reg	Registru de pagina DMA
00a0-00bf :	pic2	Programmable Interrupt Controller (PIC) 2
00c0-00df :	dma2	Controller Direct Memory Access (DMA) 1
00f0-00ff :	npu	Procesor in virgula mobila (unitate procesor numeric)
0170-0177 :	ide1	Interfata Integrated Drive Electronics 1
01f0-01f7 :	ide0	Interfata Intergrated Drive Electronics 0
0210-021f :	3c509	Placa Ethernet 3Com 3C509
02f8-02ff :	serial(auto)	Al doilea port serial (COM2)
0330-0333 :	aha1542	Controller SCSI Adaptec AHA1542
0376-0376 :	ide1	Interfata Integrated Drive Electronics 1
03c0-03df :	vga+	Controller video (placa grafica)
03f0-03f5 :	floppy	Unitate de discheta
03f6-03f6 :	ide0	Interfata Integrated Drive Electronics 0
03f7-03f7 :	floppy DIR	Unitate de discheta
03f8-03ff :	serial(auto)	Primul port serial (COM1)

Linux pe laptop-uri

Laptop-urile dispun de un grad de integrare superior celor al sistemelor desktop. Placa video, monitorul și hard-discul unui laptop sunt toate incluse într-un pachet compact. Cu alte cuvinte, componentele unui laptop nu mai pot fi combinate la fel de simplu ca în cazul sistemelor desktop, deci trebuie să vă asigurați că Linux include suport pentru toate componentele sistemului laptop.

Majoritatea calculatoarelor laptop echipate cu procesoare Intel 80386 trebuie să fie capabile a rula Linux. Dacă doriți să instalați XFree86 (X Window System), este posibil să aveți probleme, deoarece placa video (într-un laptop, circuitele plăcii video sunt incluse în placa de bază) și dispozitivul de indicare trebuie să fie suportate de XFree86.

Mai multe despre Linux pe sisteme laptop

Acest capitol reprezintă o introducere timidă în ceea ce privește rularea Linux pe sisteme laptop. Pe măsură ce utilizatorii au instalat și au rulat Linux pe o varietate de laptop-uri, experiența cumulată a comunității de utilizatori Linux pe laptop continuă să se acumuleze. O bună parte din această informație este rezumată și există pe Web. Pentru informații detaliate referitoare la instalarea și rularea sistemului Linux pe sisteme laptop, orientați-vă browserul spre următorul sit Web:

<http://www.cs.utexas.edu/users/kharker/linux-laptop/>
Acestă pagină include legături spre mult mai multe pagini Web, fiecare conținând detalii de instalare și rulare a sistemului Linux pe un anumit laptop. În particular, veți afla dacă sunt necesare anumite operații speciale pentru a determina rularea sistemelor Linux și X pe laptop.

Capitolele 10-17 ale acestei cărți tratează componente individuale, cum ar fi plăcile video și monitoarele. Această secțiune conține anumite informații referitoare la rularea sistemului Linux pe calculatoare PC laptop.

PCMCIA

Sistemele laptop dispun de un unic tip de interfață cunoscut sub numele de PCMCIA, care reprezintă abrevierea de la Personal Computer Memory Card International Association (asociația internațională pentru plăcile de memorie ale calculatoarelor personale). Versiunea Linux de pe CD-ROM-ul anexat acestei cărți include suport pentru PCMCIA. Driverelor curente PCMCIA includ suport pentru multe controllere PCMCIA frecvent folosite, precum Databook TCIC/2, Intel i82365SL, Cirrus PD67xx și Vadem VG-468.

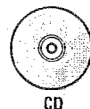


Referință
incrucisată

Examinați capitolul 17 pentru mai multe informații referitoare la suportul PCMCIA în Linux.

Gestionarea avansată a consumului de energie

O altă caracteristică specifică a sistemelor laptop o reprezintă *gestionarea consumului de energie*, care se referă la posibilitatea unui laptop de a-și suspenda activitățile pentru a economisi puterea furnizată de baterii. Calculatoarele laptop dotate cu funcția Advanced Power Management (gestionarea avansată a consumului) își pot suspenda și apoi reabilita componentele consumatoare de energie (cum sunt monitorul și unitatea de disc) și pot furniza informații referitoare la durata de viață a bateriei.



CD

Versiunea Linux de pe CD-ROM-ul anexat acestei cărți poate suporta APM, cu condiția să activați facilitățile respectivă la construcția nucleului (vezi capitolul 2 pentru informații referitoare la generarea nucleului).

Puteți afla mai multe despre suportul pentru APM sub Linux prin vizitarea sitului Web <http://www.cs.utexas.edu/users/kharker/linux-laptop/apm.html>.

Sunet pe laptop-uri

Multe laptop-uri moderne vin cu sunet încorporat. Utilizarea funcțiilor de sunet sub Linux este un proces simplu, cu condiția să vă puteți da seama care este tipul plăcii de sunet inclusă în calculator. În capitolul 12 vom discuta în detaliu despre plăcile de sunet.

Tipul plăcilor de sunet poate varia chiar și între modelele de laptop de la același vânzător. De exemplu, sistemul NEC Versa P include o placă de sunet compatibilă Sound Blaster Pro. Pe de altă parte, sistemul NEC Versa M are o placă de sunet Microsoft Sound System (MSS), care nu este compatibil Sound Blaster.

Așa cum s-a văzut în capitolul 12, configurarea specifică a plăcii de sunet Sound Blaster Pro sunt: adresă I/O 0x220 (adică 220 în notație hexazecimală; prefixul 0x semnifică hexazecimal în limbajul de programare C), IRQ 5 și DMA 1. Astfel, pe NEC Versa P, trebuie să configurați nucleul Linux cu un Sound Blaster Pro la adresa 0x220, IRQ 5 și DMA 1. Sistemul Texas Instruments TravelMate 4000M include o placă de sunet încorporată, care necesită setări similare.

Pe sistemul NEC Versa M, încercați Microsoft Sound System (MSS) la adresa I/O 0x530, IRQ 9 și DMA 3. Ca de obicei, trebuie să vă consultați documentația sistemului dumneavoastră pentru a afla detalii despre producătorul și modelul plăcii de sunet înainte de a configura suportul pentru sunet.

Așadar este posibilă configurarea sistemului Linux pentru a include suport de sunet într-un mod similar cu configurarea unui calculator de birou. Prima etapă constă în identificarea

tipului plăcii de sunet de care dispune laptop-ul. Desigur, unele laptop-uri pur și simplu nu dispun de posibilități de redare a sunetelor, deci acest paragraf este perfect inutil pentru anumiți posesori de laptop-uri.

X pe laptop-uri

Utilizatorii au semnalat rularea cu succes a sistemului Linux în paralel cu XFree86 pe multe sisteme laptop echipate cu procesoare 386, 486 și Pentium. Până acum, pentru a rula X pe un laptop, trebuia să acceptați serverul VGA standard pe 16 culori (XF86_VGA16), deoarece XFree86 nu includea suport pentru majoritatea cipseturilor video frecvent folosite în sistemele laptop. Multe laptop-uri includ suport pentru ecrane cu cristale lichide (LCD) la rezoluții de 640×480 și 256 de culori, dar serverul VGA pe 16 culori nu exploatează posibilitatea de utilizare a 256 de culori. Astfel, trebuie să puteți rula serverul X SuperVGA (XF86_SVGA) pentru a folosi 256 de culori pe un laptop.

Pentru a afla mai multe detalii despre rularea sistemului X pe laptop-uri, orientați-vă browserul Web la adresa <http://www.castle.net/X-notebook>. La această pagină puteți găsi și o legătură spre o listă de cipseturi video folosite în sistemele laptop. Majoritatea acestor sisteme folosesc următoarele tipuri de cipseturi video:

- Western Digital, seria WD90C24
- Chips & Technologies, seria 655xx
- Cirrus Logic CL-GD6253, CL-GD6440 și CL-GD7543

Serverul XF86_SVGA din XFree86 3.3.2 include suport pentru următoarele cipseturi:

- Western Digital, seria WD90C24
- Chips & Technologies, cipseturile 65520, 65525, 65530, 65535, 65540, 65545, 65546, 65550 și 65554
- Cirrus Logic, cipseturile CL-GD6225, CL-GD6235, CL-GD6420, CL-GD6440, CL-GD754x și CL-GD7555

XFree 3.3.1 include suport pentru majoritatea cipseturilor Cirrus Logic. Cipsetul Chips & Technologies 65550 este popular pe laptop-uri de nivel înalt, cum ar fi NEC Versa 6200MMX. Acest cipset include modul Super VGA la rezoluție de 1280×768, cu 64000 de culori.



Sugestie

Dacă intenționați să rulați X pe un laptop, cea mai bună opțiune o constituie sistemele echipate cu cipseturile WD90C24, 65550 sau CL-GD6440. Aceste cipseturi sunt de dată recentă și XFree86 3.3.1 le tolerează bine.

Rezumat

Linux este mult mai dependent de detaliile componentelor hardware ale calculatorului dumneavoastră decât DOS. Pentru a instala și a folosi Linux, trebuie să vă asigurați că acesta include suport pentru diferitele echipamente ale sistemului dumneavoastră de calcul. Capitolul de față se concentrează asupra compatibilității sistemului Linux cu cea mai importantă componentă a unui calculator, placa de bază, care conține procesorul și magistralele, acolo unde se conectează echipamentele periferice, cum sunt hard-discul, placa video și unitatea CD-ROM.

Din lectura acestui capitol, ați învățat următoarele:

- ▶ Linux a fost conceput pentru procesorul Intel 80386 și include suport pentru toate variantele de 386, 486, Pentium și Pentium II, inclusiv pentru toate procesoarele compatibile, cum sunt cele produse de Cyrix și AMD. De asemenea, Linux a fost portat pe alte procesoare, precum DEC Alpha AXP, Sun SPARC, HP-PA RISC, MIPS și PowerPC.
- ▶ O *magistrală* este un set standard de cablaje prin intermediul căruia un echipament periferic se conectează la placa de bază a calculatorului. Există numeroase tipuri de magistrală: ISA, EISA, VLB, MCA și PCI. Linux include suport pentru toate aceste tipuri de magistrală. Magistrala PCI reprezintă viitorul standard de magistrală de înaltă performanță pentru noile sisteme Pentium PC.
- ▶ Linux include sistemul de fișiere `/proc` ca o modalitate convenabilă de accesare a informațiilor de sistem. Puteți afla multe detalii despre sistemul dumneavoastră prin examinarea fișierelor din sistemul `/proc`. De exemplu, fișierul `/proc/devices` prezintă o listă cu toate dispozitivele existente în sistemul dumneavoastră, iar `/proc/pci` afișează dispozitivele PCI dintr-un sistem PCI.
- ▶ Laptop-urile reprezintă o categorie specială de calculatoare, deoarece toate componentele unui laptop – de la hard-disc la placa video – sunt puternic integrate. Linux rulează pe mai multe tipuri de sisteme laptop. Problema principală o constituie suportul pentru anumite caracteristici ale sistemelor laptop, precum gestionarea avansată a consumului de energie (Advanced Power Management – APM) și interfetele de tip PCMCIA sau PC Card. O altă problemă rezidă în rularea sistemului X pe un laptop, deoarece cipseturile video folosite pe astfel de calculatoare nu sunt suportate de XFree86 la fel de bine ca plăcile video ale calculatoarelor de birou. Cu toate acestea, Linux și X rulează pe o mare varietate de sisteme laptop.

Capitolul 10 Plăci video și monitoare

În acest capitol:

- ▶ Veți examina diferite tipuri de plăci video și monitoare
- ▶ Veți selecta un program server X pentru placa video și monitorul dumneavoastră
- ▶ Veți specifica parametrii de configurare ai plăcii video și ai monitorului pentru XFree86
- ▶ Veți examina anumite informații de configurare pentru diferite tipuri de plăci video

Când instalați Linux pe un PC, placa video și monitorul nu contează prea mult dacă lucrați numai cu text. Detaliile specifice ale plăcii video și ale monitorului devin importante dacă doriți să folosiți interfața grafică furnizată de X Window System, pe scurt X. Dacă doriți să beneficiați de simplitatea în utilizare furnizată de interfețele grafice, este de dorit să instalați X. Când instalați X, aveți nevoie de informații detaliate despre placa video și monitorul dumneavoastră pentru a pune X la treabă pe calculatorul dumneavoastră.

Acest capitol descrie diferite tipuri de plăci video și monitoare frecvent folosite, atributele acestora și detaliile necesare utilizării sistemului X cu placa video și monitorul respective. De asemenea, veți învăța să specificați unii parametri ai plăcii video și ai monitorului necesari pentru configurarea sistemului X pentru Linux.

Plăci video și monitoare

Placa video sau adaptorul grafic conține componentele electronice care controlează monitorul. În majoritatea sistemelor, placa video se prezintă sub forma unei plăci de circuite care se conectează într-un slot de pe placa de bază. Pe multe sisteme de producție recentă, placa de bază conține toate cipseturile grafice necesare.

Display cu baleiaj continuu

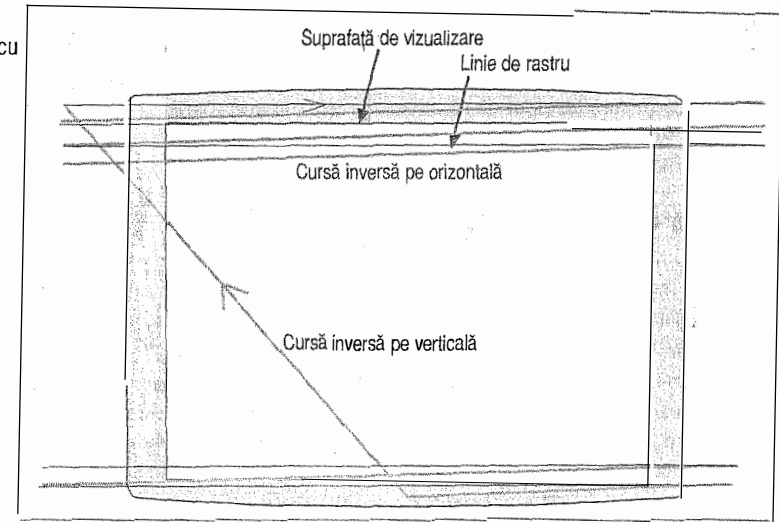
Toate plăcile video operează pe același principiu: stochează o imagine în memoria video (cunoscută și sub numele de *RAM video* sau *VRAM*) și generează semnalele adecvate necesare pentru afișarea imaginii pe ecranul monitorului.

Monitorul este echipamentul fizic care conține ecranul de afișare, acolo unde apar ieșirile grafice sau sub formă de text. *Ecranul de afișare* este, de obicei, un tub de sticlă acoperit cu fosfor pe care un fascicul de electroni urmărește imaginea de ieșire. Pe sisteme laptop, ecranul de afișare este un ecran cu cristale lichide (Liquid Crystal Display – LCD). Multe sisteme laptop pretențioase folosesc ecrane de afișare cu matrice activă.

Imaginea care apare pe monitor este alcătuită dintr-un număr mare de linii orizontale, cunoscute sub numele de *linii de rastru*. Un fascicul de electroni din monitor generează liniile de rastru prin baleierea înainte și înapoi a unui ecran acoperit cu fosfor, așa cum se vede în figura 10.1.

Figura 10-1

Un display tipic cu baleiaj continuu.



Fosforul de pe ecran strălucește direct proporțional cu intensitatea fasciculului de electroni. Punctul strălucitor de pe ecran reprezintă un element de imagine sau *pixel*. Astfel, o linie de imagine este generată prin controlul intensității fasciculului, în timp ce acesta baleiază ecranul. Peste un timp, fosforul își diminuează intensitatea, dar dacă liniile sunt trasate în mod repetat, persistența vederii noastre creează iluzia unei imagini stabile. Majoritatea monitoarelor de PC trasează un ecran întreg de linii de rastru de 50-90 de ori pe secundă.

Așa cum se vede în figura 10.1, fasciculul de electroni baleiază o suprafață mai mare decât suprafața efectivă de vizualizare, dar fasciculul de electroni este activ numai când se află în suprafața vizibilă. De asemenea, după ce ajunge la capătul unei linii, fasciculul trebuie să revină la începutul următoarei linii de rastru. Această componentă a mișcării fasciculului este cunoscută sub numele de *cursă inversă pe orizontală*. Similar, când fasciculul ajunge la capătul inferior al ecranului, trebuie să revină la prima linie pentru a începe un alt ciclu de desenare. Această perioadă este cunoscută sub numele de *cursă inversă pe verticală*. Intensitatea fasciculului se reduce în timpul curselor inverse pe orizontală și verticală, motiv pentru care aceste linii nu apar pe ecran.

Placa video generează semnalele necesare pentru parcurgerea ecranului de către fasciculul de electroni și pentru reîmprospătarea ecranului cu o viteză mare. Există două tipuri de reîmprospătare: *întreșută* (*interlaced*) și *neîntreșută* (*non-interlaced*). În modul întreșut, fiecare ecran este desenat în două etape. Mai întâi, fasciculul de electroni parcurge ecranul, desenând toate liniile impare de rastru. Apoi, fasciculul revine la începutul ecranului și desenează toate liniile pare de rastru. Modul întreșut este folosit în televiziune. Modul neîntreșut implică trasarea tuturor liniilor de rastru într-o singură etapă. Majoritatea plăcilor video reîmprospătează ecranul de 60 de ori pe secundă (sau mai mult) și folosesc modul neîntreșut.

Display color

Ecranele cu display color reprezintă orice culoare sub forma unei combinații a celor trei culori primare: roșu (R), verde (green-G) și albastru (blue-B). Majoritatea monitoarelor color folosesc trei fascicule de electroni, câte unul pentru fiecare culoare primară.

Ecranul unui display în culori folosește un șablon triunghiular de puncte fosforescente de culoare roșie, verde și albastră. Fiecare punct strălucește mai intens atunci când este atins de fasciculul de electroni. Un ecran de metal perforat, cunoscut sub numele de *mască perforată*, garantează că fiecare fascicul de electroni lovește punctul fosforescent de culoarea adecvată. Placa video variază intensitatea fasciculelor de electroni roșii, verzi și albastre, determinând astfel afișarea mai multor culori.

Paleta de culori și rezoluția

De regulă, o placă video permite o paletă de 256 de culori, caz în care culoarea fiecărui pixel este stocată într-o valoare de 8 biți. Culoarea reală (în componente RGB) care corespunde unei valori de 8 biți a unui pixel este determinată prin consultarea unei tabeli de căutare a culorilor sau a unei *hărți de culoare*.

Plăcile video de înaltă performanță permit stocarea a trei baiți pentru fiecare pixel, deci valoarea fiecărui pixel poate specifica în mod direct componentele RGB care determină culoarea pixelului respectiv. Aceste așa-numite plăci video pe 24 de biți asigură un afișaj „true color”, dar necesită o cantitate de memorie video superioară pentru stocarea întregii imagini.



Observație

Rezoluția unui ecran de display este exprimată sub forma numărului de puncte (pixeli) vizibile de-a lungul unei linii de rastru și a numărului total de linii vizibile. O rezoluție comună este 640 de puncte pe 480 de linii verticale, care se exprimă sub forma 640×480. Alte rezoluții de ecran sunt 800×600, 1024×768 și 1280×1024.

Memoria RAM video

Placa video memorează conținutul pixelilor în memoria cu acces aleator (RAM) cunoscută sub numele de memorie RAM video. Numărul de culori și rezoluția ecranului suportate de o placă video depind de cantitatea de memorie RAM video. De exemplu, pentru a stoca conținutul informațional al unui ecran ce poate afișa 256 de culori la o rezoluție de 1024×768, placa video necesită $1024 \times 768 = 786\,432$ baiți de memorie RAM video (deoarece valoarea de 8 biți a unui pixel reprezintă 256 de culori și 1 bait = 8 bait). Pe de altă parte, pentru a afișa culori pe 24 de biți la aceeași rezoluție, placa video necesită de trei ori mai multă memorie RAM video, adică $3 \times 786\,432 = 2\,359\,296$ baiți, sau aproximativ 2,3 MB de memorie RAM video (deoarece 24 biți = 3 baiți).

Plăcile video obișnuite dispun de 1 MB, 2 MB sau 3 MB de memorie RAM video. O placă video cu 1 MB de memorie RAM video poate manipula cu ușurință afișarea a 256 de culori la o rezoluție de 1024×768.

Frecvența de desenare

Veți întâlni termenul de *frecvență de desenare* (*dot clock*) la configurarea unei plăci video pentru a rula cu XFree86 (adică sistemul X Window System). *Frecvența de desenare* desemnează viteza cu care placa video poate traversa liniile de rastru care alcătuiesc un ecran de afișare complet. Valoarea frecvenței de desenare este exprimată sub forma numărului de puncte desenate într-o secundă.



Pentru a vă face o idee aproximativă cu privire la frecvența de desenare, să considerăm un ecran la 604×480, care conține $604 \times 480 = 307\,200$ de puncte (vizibile). Pentru a genera iluzia unei imagini stabile, aceste puncte trebuie redeseinate de cel puțin 72 de ori pe secundă. Astfel, placa video are de desenat

$640 \times 480 \times 72 = 22\,118\,400$ de puncte pe secundă. Ca valoare aproximativă, această viteză ajunge la 22 milioane de puncte pe secundă, adică o frecvență de desenare de 22 MHz (1MHz = 1 milion de ori pe secundă).

În realitate, pentru afișarea unui ecran de 640×480 redeseinat la 72 Hz (adică de 72 de ori pe secundă) este necesară o frecvență de desenare mai mare, deoarece fasciculul de electroni nu poate reveni instantaneu la capătul liniei. Așa cum se vede în figura 10.1, fasciculul trebuie să traverseze o linie de baleiere situată dincolo de numărul de puncte vizibile înainte de a putea reveni la începutul liniei următoare. Astfel, frecvența de desenare necesară pentru o rezoluție de 640×480 la 72 Hz este de 25,2 MHz.

După cum v-ați dat seama până acum, un ecran cu o rezoluție și mai mare necesită frecvențe de afișare superioare. Un ecran la 1024×768 și rată de reimprospătare de 72 Hz are nevoie de afișarea unui număr minim de $1024 \times 768 \times 72 = 56\,623\,104$ puncte pe secundă. Astfel, puteți cere ca frecvența de desenare pentru un ecran de 1024×768 să fie undeva deasupra valorii de 56,6 MHz.

Plăcile video mai vechi suportă un set fixat de frecvențe de desenare, dar plăcile de fabricație mai recentă includ o frecvență de desenare programabilă. Când o placă video are o frecvență de desenare programabilă, serverul X poate seta placa video pentru a opera cu orice frecvență de desenare care se încadrează într-un domeniu de valori acceptabile. Pentru o placă video cu frecvență de desenare variabilă, totuși, este necesară specificarea chip-ului care controlează frecvența de desenare (cunoscut sub numele de *chip de ceas* – *clock chip*).

Importanța plăcii video și a monitorului

Linux poate lucra cu orice combinație de monitor și placă video în mod text. Dacă o placă video și un monitor funcționează sub MS-DOS sau Windows, combinația va fi funcțională și sub Linux, în mod text. Situația se schimbă la instalarea sistemului XFree86, adică X Window System pentru Linux. Deoarece XFree86 controlează placa video în mod direct (MS-DOS folosește moduri standard predefinite pentru placa video), configurarea XFree86 pentru a lucra cu o anumită placă video necesită timp.

Monitorul este de asemenea important pentru XFree86. Semnalele electrice de la placa video controlează monitorul, deci monitorul trebuie să fie compatibil cu placa video. Ieșirea pe monitor se datorează unui fascicul de electroni care se deplasează cu viteză mare, controlat de semnalele emise de placa video. Compatibilitatea unui monitor cu placa video este corelată cu viteza cu care placa video încearcă să deplaseze fasciculul de electroni pe ecran.

Rezoluțiile suportate de combinația placă video-monitor depind de cantitatea de memorie video de pe placă și de viteza cu care se poate deplasa fasciculul de electroni al monitorului.

XFree86 își preia informația necesară despre monitor și placa video dintr-un fișier special, denumit `XFree86Config`, care se află plasat de obicei în directorul `/etc/X11`.

Selectarea programului server X

XFree86 conține numeroase programe server X, fiecare fiind conceput pentru un anumit set de plăci video. Pentru a selecta serverul X ce urmează a fi folosit, trebuie să cunoașteți tipul de placă video de care dispune calculatorul dumneavoastră. Pentru sisteme noi, puteți afla această informație interesându-vă la producătorul calculatorului. În mod ideal, obțineți informația despre placa video înainte de a comanda calculatorul, deci puteți fi sigur că Linux include suport pentru placa video respectivă.

Dacă dispuneți de un calculator mai vechi, determinarea tipului de placă video poate fi dificilă. Ca informație minimală, trebuie să cunoașteți tipul de cipset video folosit de placa video a sistemului dumneavoastră. Este și mai bine dacă știți producătorul și modelul plăcii video.



Sugestie

Dacă nu aveți nici o informație despre placa video a sistemului dumneavoastră, puteți totuși să rulați X. Presupunând că sistemul dumneavoastră este destul de recent, este posibil ca placa sa video să emuleze placa video originală Video Graphics Array (VGA) pe 16 culori. Dacă ați rulat Microsoft Windows cu un driver VGA, de exemplu, atunci știți că placa dumneavoastră video funcționează ca un VGA generic. XFree86 include un server X denumit XF86_VGA16, conceput special pentru modul video generic 640×480 pe 16 culori. Deși este posibil ca modul VGA pe 16 culori să nu exploateze la maximum capacitățile plăcii video ale sistemului dumneavoastră, cel puțin aveți posibilitatea de a folosi X.

XFree86 versiunea 3.3.2 include următoarele servere X (fiecare se prezintă sub forma unui program executabil separat):

- XF86_8514 pentru plăci video bazate pe cipsetul video IBM 8514/A.
- XF86_AGX pentru plăci video bazate pe cipsetul video AGX.
- XF86_Mach32 pentru plăci video bazate pe cipsetul video ATI Mach32.
- XF86_Mach64 pentru plăci video bazate pe cipsetul video ATI Mach64.
- XF86_Mach8 pentru plăci video bazate pe cipsetul video ATI Mach8.
- XF86_Mono pentru plăci video care operează în modul video monocrom sau modul VGA în două culori.
- XF86_P9000 pentru plăci video bazate pe cipsetul video Weitek P9000. Plăcile video Diamond Viper și Orchid P9000 folosesc acest cipset.
- XF86_S3 pentru plăci video bazate pe cipsetul video S3. Rețineți că nu sunt suportate toate plăcile video bazate pe S3. Trebuie să dispuneți de o placă video suportată de serverul respectiv.
- XF86_SVGA pentru plăci video care operează în modul Super VGA. În această categorie este inclusă o mare varietate de plăci video, ce folosesc o gamă diversă de cipseturi.
- XF86_VGA16 pentru o placă video care operează în modul VGA pe 16 culori. Puteți folosi acest server X cu majoritatea plăcilor video, inclusiv cu plăcile VGA mai vechi și EGA (Enhanced Graphics Adapter).
- XF86_W32 pentru plăci video bazate pe cipsetul video Tseng ET4000/W32.

Această listă de programe server poate fi mai mare în versiunile ulterioare ale sistemului XFree86. Pentru ultimele informații privind anumite tipuri de plăci video, examinați fișierul README din directorul /usr/X11/lib/X11/doc.



Referință încrucișată

Așa cum s-a arătat în capitolul 4, trebuie să rulați programul utilitar xfb8config pentru a seta XFree86 pe sistemul dumneavoastră. La momentul respectiv, când specificați o placă video, programul xfb8config vă permite selectarea unui program server X. După ce ați selectat un anumit server X, programul xfb8config configurează /usr/X11R6/bin/X ca legătură simbolică la fișierul binar al serverului X efectiv. De exemplu, pentru a putea folosi serverul XF86_S3, utilitarul xfb8config execută următoarea comandă:

```
ln -s /usr/X11R6/bin/XF86_S3 /usr/X11R6/bin/X
```

Când lansați X folosind scriptul startx sau folosind xdm (așa cum s-a explicat în capitolul 5), serverul X care este legat în mod simbolic la /usr/X11R6/bin/X este cel care va fi lansat în execuție. Când serverul X este lansat, acesta citește conținutul fișierului XF86Config (din directorul /etc/X11) pentru a configura placa video și monitorul.

Din nou despre fișierul XF86Config

Când serverul X se lansează, acesta găsește informații despre placa video, monitor, tastatură și mouse din fișierul XF86Config (plasat, de regulă, în directorul etc/X11). În capitolul 4 am examinat fișierul /etc/X11/XF86Config, dar nu am descris în totalitate modul de completare a unor secțiuni importante ale fișierului respectiv. Această secțiune va examina fișierul XF86Config sub aspectul furnizării informațiilor pentru anumite plăci video și monitoare.

Așa cum s-a văzut în capitolul 4, XF86Config este alcătuit din mai multe secțiuni. Dintre acestea, trei sunt cele care necesită date despre placa video și despre monitor:

- Secțiunea Screen (ecran) combină o placă video și un monitor cu modulele video care urmează a fi folosite într-un anumit server X. De regulă, XF86Config conține mai multe secțiuni Screen, una pentru fiecare tip de server X.
 - Secțiunea Device (dispozitiv) descrie caracteristicile unei plăci video (cunoscută și sub numele de *dispozitiv grafic* – *graphic device*). Fișierul de configurare poate conține numeroase secțiuni Device.
 - Secțiunea Monitor prezintă specificațiile tehnice ale monitorului (cum ar fi ratele de sincronizare pe verticală și pe orizontală) și include o listă cu modulele video suportate de monitor.
- Așa cum s-a explicat în capitolul 4, puteți crea fișierul XF86Config prin rularea programului utilitar xfb8config. Dacă aveți la îndemână specificațiile tehnice ale monitorului și placa video apare în lista afișată de xfb8config, fișierul XF86Config poate funcționa fără modificări. De obicei, problemele apar atunci când încercați să folosiți posibilitățile complete ale plăcilor video avansate. În secțiunile următoare ale acestui capitol este descris modul de utilizare a sistemului XFree86 cu anumite plăci video de natură mai complexă.

Secțiunea Screen

Serverul X decide asupra setărilor plăcii video și a monitorului din secțiunea Screen destinată serverului respectiv. Serverele X sunt grupate în patru categorii, denumite după cum urmează:

- vga2 se referă la o combinație placă video-monitor care operează în mod bicolor.
- vga16 se referă la modul VGA standard pe 16 culori.
- svga se referă la modul Super VGA (care desemna inițial modul 640×480 pe 256 de culori, dar serverul X Super VGA include suport pentru aproape toate plăcile video și chiar pentru acceleratoarele grafice de pe multe plăci video).
- accel se referă la toate serverele X accelerate. (Alegerea serverului X depinde de cipsetul folosit. Serverul se selectează la rularea programului utilitar xfb8config.)

Secțiunea Screen specifică serverul X care poate folosi secțiunea respectivă, precum și numele unui device (placă video) și al unui Monitor. În continuare este prezentată o secțiune Screen destinată unui server X accelerat:

```
# Screen for accelerated servers (S3, Mach8, 8514, P9000, AGX, W32, Mach64)
Section "Screen"
    Driver      "accel"
    Device      "Number Nine GXE64 with S3-Trio64"
```

```

Monitor "Dell VS15X"
Subsection "Display"
    Depth 8
    Modes "640x480" "800x600" "1024x768"
    ViewPort 0 0
    Virtual 1024 768
EndSubsection
Subsection "Display"
    Depth 16
    Modes "640x480" "800x600"
    ViewPort 0 0
    Virtual 800 600
EndSubsection
Subsection "Display"
    Depth 32
    Modes "640x400"
    ViewPort 0 0
    Virtual 640 400
EndSubsection
EndSection

```

Liniile de comentariu încep cu simbolul diez (#). Definiția secțiunii este inclusă în blocul Section...EndSection.

Linia Driver identifică serverul X care trebuie să folosească această secțiune Screen. În continuarea numelor Device și Monitor există numeroase secțiuni Display. Fiecare subsecțiune Display se aplică unei anumite profunzimi (Depth), adică numărul de biți de stocare per pixel, care mai determină și numărul de culori care pot fi afișate la un moment dat.



Serverul X folosește în mod automat prima subsecțiune Display în definiția Screen, dar puteți lansa serverul cu opțiuni în linie de comandă care specifică o altă profunzime. Dacă lansați serverul X folosind comanda startx, puteți folosi o profunzime de 24 prin emiterea următoarei comenzi:

```
startx - -bpp 24
```

Opțiunea -bpp reprezintă prescurtarea de la biți per pixel, similară cu Depth. Selectarea unei valori pentru Depth are efect cu condiția ca placa video și monitorul să poată suporta valoarea respectivă.

Linia Modes din subsecțiunea Display prezintă numele modurilor video pe care le pot suporta monitorul și placa video. Numele acestor moduri video apar în secțiunea Monitor a fișierului XF86Config.

Dacă placa video dispune de mai multă memorie decât este necesar pentru a reține informația aferentă tuturor pixelilor vizibili într-un anumit mod, serverul X poate folosi surplusul de memorie pentru a afișa un tablou de pixeli cu mult mai mare decât 640x480 sau 1024x768, care pot fi specificate de un mod video standard. Cu alte cuvinte, utilizatorul are la dispoziție un ecran virtual mare, din care poate selecta o suprafață de vizualizare mai mică. Linia Virtual indică dimensiunea acestui ecran virtual, în timp ce ViewPort specifică partea din ecranul virtual care este mapată pe afișajul fizic.

Secțiunea Device

Secțiunea Device a fișierului XF86Config pune la dispoziția utilizatorului informații despre placa video. Pentru modul VGA standard, XF86Config dispune de o secțiune Device care se prezintă astfel:

```

# Standard VGA Device:
Section "Device"

```

```

Identifier "Generic VGA"
VendorName "Unknown"
BoardName "Unknown"
Chipset "generic"
# VideoRam 256
# Clocks 25.2 28.3
EndSection

```

Fiecare linie a acestei secțiuni pune la dispoziția utilizatorului informații despre placa video. În acest caz, linia Identifier indică tipul plăcii video. Acest identificator este folosit în secțiunea Screen pentru a desemna un anumit tip de placă video.

VendorName, BoardName și Chipset prezintă mai multe detalii privind placa video și cipsetul video. Pentru modul VGA standard, cele trei linii de mai sus nu prezintă importanță. De asemenea, nu este nevoie să specificați cantitatea de memorie RAM video (de pe linia VideoRam) sau frecvențele de desenare permise (din linia Clocks). Ca atare, aceste linii sunt transformate în comentarii prin inserția simbolului diez (#) la început.

Pentru multe plăci video care nu dispun de o frecvență de desenare programabilă, cea mai importantă linie din secțiunea Device este linia Clocks. Valorile din această linie indică frecvențele de desenare suportate de placa video.

Dacă doriți, programul xf86config poate rula serverul X cu o opțiune specială (-probeonly), determină frecvențele de desenare suportate de placa video și completează linia Clocks din secțiunea Device. Tot ce aveți de făcut este să răspundeți afirmativ când programul xf86config întreabă dacă doriți să rulați în mod probeonly. La execuția comenzii x -probeonly, serverul X consultă fișierul XF86Config. Serverul X nu mai testează frecvențele de desenare dacă fișierul conține deja o linie Clocks. De asemenea, fișierul de configurare trebuie să dispună de secțiunile Keyboard și Pointer corect specificate pentru a ajunge la testarea plăcii video. În particular, dacă există o eroare de specificare a tipului de mouse, x -probeonly eșuează.

În afară de linia Clocks, puteți specifica unul sau mai multe indiciatori destinate serverului X pentru tipul de placă video specificat în secțiunea Device.

Secțiunea Monitor

Secțiunea Monitor prezintă specificațiile tehnice ale monitorului: frecvența de sincronizare pe orizontală (*horizontal sync*) și rata de sincronizare pe verticală. Puteți obține aceste valori din manualul monitorului dumneavoastră.



Semnalul de frecvență de sincronizare pe orizontală survine la sfârșitul fiecărei linii de rastru; acest semnal deplasează fasciculul de electroni de la sfârșitul unei linii de rastru la începutul următoarei linii. Frecvența de sincronizare pe orizontală reprezintă, în esență, numărul de ori pe secundă cu care monitorul poate urmări o linie de rastru pe ecran. Dacă monitorul poate afișa 480 de linii (la o rezoluție de 640x480, de exemplu) și poate redesea ecranul de 72 de ori pe secundă (rată de sincronizare pe verticală de 72 Hz), frecvența de sincronizare pe orizontală este de minimum $480 \times 72 = 34560$ de ori pe secundă $= 34.560 \text{ Hz} = 34,56 \text{ kHz}$. Valoarea reală este superioară, deoarece monitorul trebuie întotdeauna să afișeze mai multe linii decât cele care sunt vizibile.



Atenție!

Trebuie să fiți atent la valorile frecvențelor de sincronizare pe verticală și pe orizontală, deoarece serverul X folosește aceste valori pentru a selecta semnalele trimise de la placa video. Un monitor poate suferi deteriorări fizice dacă placa video trimite semnale care depășesc specificațiile monitorului.

Când folosiți programul `xf86config` pentru a crea fișierul `XF86Config`, programul `xf86config` vă oferă opțiunea de selectare a frecvențelor de sincronizare pe verticală și pe orizontală dintr-o listă generală, cum ar fi un monitor VGA standard sau super VGA. Dacă nu dispuneți de informații despre monitor, puteți începe cu o setare conservatoare pentru sincronizarea pe orizontală, cum ar fi valoarea aferentă unui monitor VGA standard. (Nu este necesar să introduceți valoarea exactă a frecvenței de sincronizare pe orizontală; pur și simplu selectați tipul monitorului.)

Cele mai importante intrări din secțiunea Monitor o reprezintă liniile Modeline, care prezintă o listă a modurilor video adecvate pentru a fi utilizate cu monitorul. De regulă, majoritatea utilizatorilor începători au probleme cu completarea liniilor Modeline. De fapt, nu aveți prea multe de făcut cu liniile Modeline; programul utilitar `xf86config` generează un fișier `XF86Config` care include un set standard de sincronizări de mod. Dacă se specifică frecvențele de sincronizare pe verticală și pe orizontală (*rata de sincronizare pe verticală* indică numărul de desenări corecte ale ecranului într-o secundă), serverul X selectează în mod automat cel mai bun mod video suportabil de placa video și de monitor la o rezoluție dată.

În continuare este prezentată o secțiune Monitor specifică (cu majoritatea liniilor Modeline) șterse: Section "Monitor"

```

Identifier "Dell VS15X"
VendorName "Dell"
ModelName "VS15X"

# HorizSync se masoara in kHz daca nu sunt specificate alte
# unitati.
# HorizSync poate fi o lista de valori discrete separate prin
# virgula sau o lista de domenii de valori separate prin
# virgula.
# NOTA: VALORILE PREZENTATE AICI AU VALOARE ORIENTATIVA.
# PENTRU A AFLA VALORILE CORECTE, CONSULTATI DOCUMENTATIA
# MONITORULUI DUMNEAVOASTRA

HorizSync 30-64

# HorizSync 30-64 #multisync
# HorizSync 31.5, 35.2 #frecv. de sincronizare multiple fixe
# Horizsync 15-25, 30-50 #domenii multiple de frecvente

# VertRefresh se masoara in Hz daca nu se specifica alte unitati.
# VertRefresh poate fi o lista de valori discrete separate prin virgula sau o
# lista de domenii de valori separate prin virgula.
# NOTA: VALORILE PREZENTATE AICI AU VALOARE ORIENTATIVA.
# PENTRU A AFLA VALORILE CORECTE, CONSULTATI DOCUMENTATIA
# MONITORULUI DUMNEAVOASTRA

VertRefresh 55-90

# Modurile pot fi specificate in doua formate: compact (pe o singura linie) sau
# multilinie.

# Urmatoarele specificatii sunt echivalente

# Modeline "1024xi" 45 1024 1048 1208 1264 768 776 784 817 Interlace

# Mode "1024x768i"
# DotClock 45
# HTimings 1024 1048 1208 1264
# VTimings 768 776 784 817

```

```

# Flags "Interlace"
# EndMode

# 640x400 @ 70Hz, 31.5 kHz hsync
Modeline "640x400" 25.175 640 664 760 800 400 409 411 450
# (Liniile Modeline sterse)
EndSection

```

Câmpul Identifler al secțiunii Monitor denumește monitorul respectiv; numele în cauză este folosit în secțiunea Screen pentru a specifica monitorul. Trebuie să completați liniile HorizSync și VertRefresh cu informații extrase din manualul monitorului (sau să selectați una dintre valorile tipice afișate de programul `xf86config`).

Calculul unui modeline

Deși puteți folosi modeline-urile standard generate de programul `xf86config`, este util să cunoașteți detaliile acestei linii, în cazul în care trebuie să definiți un mod unic pentru o combinație placă video – monitor.

În mod normal, un modeline se specifică pe o singură linie folosind următoarea sintaxă:

```
Modeline "nume" CLK HRES HSS HSE HTOT VRES VSS VSE VTOT flags
```

Trebuie să completați toate argumentele care apar scrise cu caractere cursive, cu excepția ultimului, care este un cuvânt cheie opțional care indică tipul modului. Câmpul `flags`, de exemplu, poate avea valoarea `Interlace` pentru un mod întrețesut (imaginea este trasată prin linii de rastu alternative) sau `DoubleScan` (fiecare linie de scan este dublată). Alte indicatoare reprezintă polaritatea semnalului de sincronizare. Valorile pot fi `+HSync`, `-HSync`, `+VSync` sau `-VSync`, în funcție de polaritățile specificate de utilizator.

Semnificațiile argumentelor incluse în Modeline sunt următoarele:

■ `"nume"` este numele modului respectiv, încadrat între ghilimele duble. De regulă, rezoluția modului este folosită ca nume al acestuia. Astfel, veți vedea nume de moduri, precum `"640x480"` sau `"1024x768"`. Aceste nume de moduri se folosesc în subsecțiunea Display a secțiunii Screen.

■ `CLK` este frecvența de desenare de folosit pentru acest mod. Pentru o placă video cu un set fixat de frecvențe de desenare, valoarea trebuie să fie una din cele specificate în linia `Clocks` a secțiunii Device din fișierul `XF86Config`.

■ `HRES HSS HSE HTOT` reprezintă parametrii de sincronizare pe orizontală. `HRES` este rezoluția pe orizontală, exprimată sub forma numărului de pixeli vizibili pe o linie de rastu. Așa cum se vede în figura 10.1, numărul efectiv de pixeli pe o linie de rastu depășește numărul pixelilor vizibili. `HTOT` este numărul total de pixeli pe o linie. `HSS` este numărul de ordine al pixelului de început al semnalului de sincronizare pe orizontală, iar `HSE` este punctul în care acest semnal se încheie. Semnalul de sincronizare pe orizontală deplasează fasciculul de electroni de la o linie la alta. Pentru modul video 640x480, cei patru parametri pot avea valorile 640 680 720 864. Conform acestei secvențe de numere, pe o linie de rastu se află 864 de pixeli, din care numai 640 sunt vizibili. Semnalul de sincronizare pe orizontală începe la pixelul 680 și se încheie la pixelul 864.

■ `VRES VSS VSE VTOT` reprezintă parametrii de sincronizare pe verticală. `VRES` este rezoluția pe verticală, exprimată sub forma numărului de pixeli vizibili pe o linie de rastu. Așa cum se vede în figura 10.1, numărul efectiv de pixeli pe o linie de rastu depășește numărul pixelilor vizibili. `VTOT` este numărul total de linii de rastu. `VSS` este numărul de ordine al pixelului de început al semnalului de sincronizare pe verticală, iar `HSE` este punctul în care acest semnal se

încheie. Semnalul de sincronizare pe verticală deplasează fasciculul de electroni din partea inferioară a ecranului la începutul primei linii. Pentru modul video 640x480, cei patru parametri pot avea valorile 480 488 491 521.

Din manualul de utilizare a monitorului, puteți obține doi parametri cheie: rata de reîmprospătare pe verticală (în Hz) și frecvența de sincronizare pe orizontală (în kHz). Manualul de utilizare a monitorului pune la dispoziția utilizatorului aceste valori sub forma unor domenii de valori corecte. De regulă, rata de reîmprospătare pe verticală este cuprinsă între 50 și 90 Hz, iar frecvența de sincronizare pe orizontală se poate situa între 30 și 135 kHz. În continuare sunt prezentate două ecuații care definesc relația dintre frecvența de desenare (dot clock) și unii parametri de sincronizare pe verticală și orizontală din linia Modeline:

$$\text{CLK} = \text{RR} * \text{HTOT} * \text{VTOT}$$

$$\text{CLK} = \text{HSF} * \text{HTOT}$$

În aceste ecuații, RR este o rată de reîmprospătare a ecranului situată în domeniul de existență al ratei de reîmprospătare a monitorului, iar HSF este o frecvență de baleiere pe orizontală suportată de monitor. Nu uitați să convertiți totul la o unitate de măsură comună (de exemplu, asigurați-vă că toate valorile sunt exprimate în Hz) atunci când aplicați aceste formule.

Pentru a defini un mod, puteți începe de la o rată de reîmprospătare dorită (RR), cum ar fi 72 Hz. Pentru o frecvență de desenare dată, puteți apoi calcula produsul $\text{HTOT} * \text{VTOT}$ din prima ecuație. Apoi, introduceți o valoare pentru HSF situată în domeniul frecvențelor de baleiere pe orizontală admise pentru monitor. Deoarece frecvența de desenare este deja cunoscută, puteți calcula HTOT din a doua ecuație. După ce ați determinat HTOT, puteți determina VTOT, deoarece ați calculat deja produsul acestor două mărimi.

În acest moment, cunoașteți HTOT și VTOT. Ceea ce aveți de selectat sunt argumentele HSS, HSE, VSS și VSE, de care aveți nevoie la completarea liniei Modeline. Din păcate, determinarea acestor parametri necesită iterații repetate. Puteți începe prin a selecta valorile HRES și VRES (care determină rezoluția modului). Apoi trebuie să selectați HSS și HSE care să se încadreze între HRES și HTOT, cu condiția $\text{HSE} > \text{HSS}$. Similar, VSE și VSS trebuie să fie cuprinse între VRES și VTOT, cu condiția $\text{VSE} > \text{VSS}$.

Dacă suprafața de afișare este mică sau descentrată, trebuie să modificați valorile HSS, HSE, VSS și VSE. Dat fiind că fișierul prestabilit XF86Config include deja liniile Modeline pentru multe moduri video comune, nu aveți de făcut decât să dați domeniile de existență ale ratei de reîmprospătare pe verticală, respectiv ale frecvenței de sincronizare pe orizontală pentru monitorul dumneavoastră, și să vă fixați la unul dintre modurile predefinite.

Plăci video comune

XFree86 include suport pentru o largă varietate de plăci video. Deși numele de firmă al plăcii video poate fi important, cipsetul video este elementul de maximă importanță pentru serverul X. XFree86 3.3.2 pune la dispoziția utilizatorului serverul XF86_SVGA, care include suport pentru plăci video bazate pe următoarele cipseturi:

- Tseng ET3000, ET4000ActiveX, ET4000/W32 și ET6000
- Western Digital/Paradise PVGA1
- Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C30, WD90C31 și WD90C33
- Genoa GVGA

- Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420, TGUI9420DGi, TGUI9430DGi, TGUI9440AGi, TGUI9660XGi și TGUI9680
- ATI 18800, 18800-1, 28800-2, 28800-4, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800GX-C, 88800GX-D, 88800GX-E, 88800GX-F, 88800CX, 264CT, 264ET, 264VT, 264VT2 și 264GT
- NCR 77C22, 77C22E și 77C22E+
- Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD5436, CLGD5440, CLGD5446, CLGD5462, CLGD5464, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6410, CLGD6412, CLGD6420 și CLGD6440
- Chips & Technologies 65520, 65530, 65540, 65545, 65546, 65548, 65550 și 65554
- Compaq AVGA
- OAK OTI067, OTI077 și OTI087
- Alliance AP6422
- Advance Logic ALG2101, ALG2228, ALG2301, ALG2302, ALG2308, ALG2401
- ARK Logic ARK1000PV, ARK1000VL, ARK2000PV și ARK2000MT
- Matrox MGA2064W și Mystique
- MX MX68000 și MX680010
- NVidia/SGS Thomson NV1 și STG2000
- RealTek RTG3106
- SiS 86C201, 86C202 și 86C205
- Video 7/Headland Technologies HT216-32

Serverul XF86_SVGA folosește aceste plăci video în modul Super VGA (SVGA), de regulă la rezoluții mai mari de 1024x768, cu 256 de culori.

Pentru ultimele noutăți în materie de cipseturi video, consultați următoarele fișiere din directorul /usr/X11/lib/X11/doc:

README.ati descrie suportul serverului XF86_SVGA pentru plăcile video din seria ATI VGA Wonder.

README.cirrus descrie suportul serverului XF86_SVGA pentru cipseturile video Cirrus Logic, inclusiv suportul pentru unele posibilități grafice de accelerare. Fișierul mai descrie opțiunile referitoare la cipseturile Cirrus.

README.trident descrie suportul serverului XF86_SVGA pentru cipseturile video Trident, inclusiv toate opțiunile care influențează cipseturile Trident.

README.video7 descrie suportul serverului XF86_SVGA pentru cipsetul video Headland Technologies HT216-32.

README.wstDig descrie suportul serverului XF86_SVGA pentru cipseturile video Western Digital, inclusiv WD90C31 și WD90C33, cu posibilități grafice de accelerare.

Plăci video accelerate

Dacă doriți să beneficiați de o grafică accelerată sub X, aveți nevoie de o placă video accelerată care folosește cipseturi video speciale. Există o mulțime de asemenea cipseturi și vor apărea și

mai multe, pe măsură ce PC-urile trec la sisteme de operare grafice, precum Microsoft Windows și X Window System.

În continuare sunt prezentate unele dintre cipseturile video accelerate:

- cipseturile S3 (86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964, 86C732 sau Trio32, 86C764 sau Trio64, 86C765, 86C868, 86C968, 86C325 și 86C988) sunt folosite într-o mare varietate de plăci video de firmă, cum ar fi Diamond Stealth 24 VLB, Diamond Stealth 64, Number Nine GXE64, Number Nine GXE64 PCI și SPEA Mirage P64. Aceste cipseturi sunt suportate de serverul XF86_S3.
- cipurile S3 ViRGE (86C325), S3 ViRGE/DX (86C375), S3 ViRGE/GX (86C385) și S3 ViRGE/VX (86C988) sunt folosite în multe plăci video de producție recentă, cum sunt Diamond Stealth 3D 2000, ELSA Winner 3000, Hercules Terminator 3D sau STB Nitro 3D. Aceste cipuri S3 mai recente sunt suportate de serverul XF86_S3V.
- cipsetul Weitek P9000 este folosit la plăci video precum Diamond Viper VLB, Diamond Viper PCI și Orchid P9000. Cipsetul este suportat de serverul XF86_P9000.
- ATI Mach8, Mach32 și Mach64 sunt folosite în plăcile video ATI, cum sunt ATI Graphics Xpression Mach64. Aceste cipseturi sunt suportate de serverele XF86_Mach8, XF86_Mach32 și XF86_Mach64.
- Tseng ET4000/W32 sunt folosite de plăci video precum Hercules Dynamite Pro VLB și Mirage ET4000/W32 VLB. Aceste cipseturi sunt suportate de serverul XF86_W32.
- cipseturile și clonele IBM 8514/A sunt folosite de plăci video precum IBM 8514/A și Western Digital WD9510-AT. Aceste cipseturi sunt suportate de serverul XF86_8514.
- cipseturile IIT AGX-014, AGX-015 și AGX-016 sunt folosite în plăci video precum cele din seria Hercules Graphite. Aceste cipseturi sunt suportate de serverul XF86_AGX.
- caracteristicile de accelerare ale cipseturilor Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD6205, CLGD6215, CLGD6225 și CLGD6235 sunt folosite cu precădere în plăcile video integrate în laptop-uri. Aceste cipseturi sunt suportate direct de serverul XF86_SVGA.
- caracteristicile de accelerare ale cipseturilor Western Digital WD90C31 și WD90C33 sunt suportate direct de serverul XF86_SVGA.
- caracteristicile de accelerare ale cipsetului Oak OTI-087 sunt suportate direct de serverul XF86_SVGA.

XFree86 include un server separat pentru fiecare cipset cu accelerare, cu excepția cipseturilor Cirrus, Western Digital și Oak. Aceste cipseturi sunt suportate direct de serverul XF86_SVGA.

În următoarele secțiuni vom prezenta câteva plăci video și cipseturi video cu accelerare frecvent utilizate. Pentru ultimele noutăți în materie de cipseturi video, examinați următoarele fișiere din directorul /usr/X11/lib/X11/doc:

README.agx descrie serverul XF86_AGX, care include suport pentru o largă varietate de plăci video bazate pe cipsetul video IIT AGX.

README.oak descrie suportul serverului XF86_SVGA pentru plăcile video bazate pe cipseturile Oak Technologies Inc. OTI067, OTI077 și OTI087.

README.P9000 descrie suportul serverului XF86_SVGA pentru plăcile video bazate pe cipsetul Weitek P9000.

README.S3 descrie serverul XF86_S3 pentru o mare varietate de plăci video bazate pe cipsetul video S3.

README.S3V descrie suportul serverului XF86_S3V pentru plăcile video bazate pe cipsetul video S3 ViRGE.

README.W32 descrie suportul serverului XF86_W32 pentru plăcile video bazate pe seriile de cipseturi video ET4000/W32.

Diamond Viper și Orchid P9000

În momentul de față, există două variante ale plăcii video Diamond Viper: Viper VLB pentru magistrale VLB și Viper PCI pentru sisteme PCI. Plăcile video Diamond Viper și Orchid P9000 sunt bazate pe cipsetul video Weitek Power 9000 (P9000), suportate de serverul X cu accelerare XF86_P9000.

Serverul XF86_P9000 se așteaptă să găsească numele producătorului în linia Chipset a secțiunii Device din fișierul XF86Config. Serverul recunoaște următoarele linii Chipset:

```
Chipset "viperpci"      #Diamond Viper PCI
Chipset "vipervlb"     #Diamond Viper VLB
Chipset "orchid_p9000" #Orchid P9000
```

În trecut, Linux nu includea suport pentru plăcile Diamond Video, deoarece firma Diamond nu a furnizat în mod gratuit informații tehnice pentru placa sa video. Pentru a obține aceste informații, era necesară semnarea unui acord de confidențialitate. De la sfârșitul lui 1994, firma a furnizat informații tehnice detaliate despre plăcile sale video pentru dezvoltatorii de XFree86, fără nici un fel de restricții. Rezultatul final a fost acela că toate plăcile Diamond de producție recentă sunt în prezent suportate de sistemul XFree86.

ATI Mach8, Mach32 și Mach64

ATI Mach8 și Mach32 sunt cipseturi video accelerate compatibile IBM 8514/A. Serverele X cu accelerare XF86_Mach8 și XF86_Mach32 suportă respectiv plăcile video ATI Mach8 și Mach32.

ATI Mach64 este un cipset video mai recent, de înaltă performanță. Mai multe tipuri de plăci video ATI, cum sunt ATI Graphics Pro Turbo și ATI Graphics Xpression, folosesc cipsetul video Mach64. Multe sisteme Pentium PCI folosesc placa video ATI Graphics Xpression. Versiunea XFree86 de pe CD-ROM-ul anexat cărții include serverul XF86_Mach64, care suportă placa video Mach64.



Secret

Unele plăci video ATI, cum sunt ATI Graphics Ultra și Ultra Pro, includ un port de mouse. Acest port se conformează protocolului de mouse de magistrală Logitech. Dacă dispuneți de un mouse conectat la portul de mouse al plăcii ATI, trebuie să instalați driverul pentru mouse de magistrală Logitech.

Plăcile video S3

Cipsetul S3 este un cipset video cu accelerare frecvent folosit de multe plăci video. Tabelul 10.1 prezintă plăcile video specifice (nume de marcă) suportate de serverul XF86_S3 în versiunea XFree86 3.3.2 – cea mai recentă versiune a programului XFree86. Coloana biți per pixel (bpp) prezintă numărul BPP maxim suportat de placa respectivă. Când în coloana BPP este afișat numărul 32, de exemplu, aceasta arată că placa suportă și modurile 8- și 16-BPP.

Tabelul 10.1 mai include plăci video bazate pe cipsetul S3 ViRGE. Serverul XF86_S3 include suport pentru aceste plăci video.

Tabelul 10.1 Plăcile video bazate pe cipsetul S3 suportate de XFree86 3.3.2

Chipset S3	Biți per pixel (bpp)	Plăci video
801/805	16	Actix GE 32, Orchid Fahrenheit 1280+, STB PowerGraph X.24, JAX 8231, SPEA Mirage, STB PowerGraph X.24
805	16	Miro 10SD VLB/PCI, SPEA Mirage VLB
805	8	Diamond Stealth 24 VLB
928	16	Actix Ultra, SPEA Mercury VLB
928	32	ELSA Winner 1000 ISA/VLB/EISA, STB Pegasus VL, Number Nine GXE Level 10/11/12, Number Nine GXE Level 14/16
864	32	Actix GE 64 VLB, Miro 20SD (BIOS 1.x, BIOS 2.x, BIOS 3.x), ELSA Winner 1000 PRO VLB/PCI, ELSA Winner 1000 PRO, SPEA Mirage P64 DRAM (BIOS 4.x), Diamond Stealth 64 DRAM, Genoa Phantom 64i, Miro Crystal 20SD VLB (BIOS 3.x), Number Nine GXE64 PCI
864	16	SPEA Mirage P64 DRAM (BIOS 3.x)
964	32	Miro Crystal 20SV PCI, Diamond Stealth 64, SPEA Mercury 64, Number Nine GXE64 Pro VLB/PCI, Miro Crystal 40SV, Hercules Graphite Terminator 64
964	8	ELSA Winner 2000 PRO PCI
868	32	ELSA Winner 1000AVI, Miro Crystal 20SD PCI
968	32	ELSA Winner 2000PRO/X, Diamond Stealth 64 Video VRAM, Genoa VideoBlitz III AVI, Hercules Terminator Pro 64, STB Velocity 64V, Number Nine FX Motion 771, Diamond Stealth 64 Video 3240/3400
732 (Trio32)	32	Diamond Stealth 64 DRAM SE, toate plăcile bazate pe cipsetul Trio32
764 (Trio64)	32	SPEA Mirage P64 (BIOS 5.x), Diamond Stealth 64 DRAM, Diamond Stealth 64 Graphics 2000, Number Nine FX Vision 330, STB PowerGraph 64, toate plăcile bazate pe cipsetul Trio64
VIRGE/DX/GX/VX	32	Diamond Multimedia Stealth 3D 2000, Diamond Stealth 3D 2000, Diamond Stealth 3D 2000 PRO, Diamond Stealth 3D 3000, ELSA Winner 3000-S, Hercules Terminator 64/3D, Number Nine FX Motion 332, Orchid Tech. Fahrenheit Video 3D, STB systems Powergraph 3D, STB Nitro 3D

Deși multe plăci video folosesc același cipset S3, între ele există două diferențe semnificative:

- cipul de ceas care generează frecvențele de desenare

- un cip, cunoscut sub numele de RAMDAC, care convertește conținutul memoriei video în semnale digitale trimise la monitor



Secret

Când selectați una dintre plăcile video bazate pe cipsetul S3, nu trebuie să specificați cipul de ceas sau cipul RAMDAC. Dacă aveți de-a face cu o placă bazată pe S3 de dată recentă, al cărei nume nu apare în lista de plăci suportate, puteți determina serverul XF86_S3 sau XF86_S3V să lucreze cu placa respectivă prin specificarea cipului de ceas și a RAMDAC în secțiunea Device a fișierului XF86Config.

Majoritatea cipurilor RAMDAC sunt detectate în mod automat de serverul X, dar dacă dispuneți de o placă video cu RAMDAC AT&T 20C490, trebuie să includeți următoarea linie în secțiunea Device a fișierului XF86Config:

```
RamDac "att20c490"
```

Tabelul 10.2 prezintă cipurile de ceas suportate și linia ClockChip pe care o puteți folosi pentru a specifica cipul în secțiunea Device a fișierului XF86Config.

Tabelul 10.2 Cipuri de ceas suportate și linia ClockChip

Cip de ceas	Linia ClockChip din fișierul XF86Config
AT&T 20C409	ClockChip "att20c409"
Chrontel 8391	ClockChip "ch8391"
DCS2824-0 (compatibil Diamond ICD 2061A)	ClockChip "dcs2824"
IBM RGB 5xx	ClockChip "ibm_rgb5xx"
ICD2061A	ClockChip "icd2061a"
ICS2595	ClockChip "ics2595"
ICS5300 GENDAC (compatibil 86c708)	ClockChip "ics5300"
ICS5342 GENDAC	ClockChip "ics5342"
ICS9161A (compatibil ICD2061A)	ClockChip "9161a"
S3 86c708 GENDAC	ClockChip "s3gendac"
S3 86c716 SDAC	ClockChip "s3_sdac"
Sierra SC11412	ClockChip "sc11412"
STG 1703	ClockChip "stg1703"
TI3025	ClockChip "ti3025"
TI3026	ClockChip "ti3026"

Servere X comerciale pentru XFree86

Dacă XFree86 nu include suport pentru o placă video cu performanțe avansate, va trebui să aveți în vedere un server comercial. Prețul acestora este cuprins între 100 și 200 de dolari, dar veți beneficia de un server X care suportă mai multe plăci decât XFree86. Pentru plăci deja suportate de XFree86, versiunea comercială poate furniza performanțe superioare decât cele obținute de XFree86.

În continuare sunt prezentate două versiuni populare de servere X:

- Metro-X de la Metro Link, Inc. Pentru informații, trimiteți e-mail la adresa sales@metrolink.com sau vizitați situl Web al firmei la adresa <http://www.metrolink.com/>.
- Accelerated-X de la Xi Graphics, Inc. Pentru informații, trimiteți e-mail la sales@xig.com sau încercați la situl Web al firmei, la adresa <http://www.xig.com/>.

În continuare prezentăm o listă a plăcilor video suportate de serverele X comerciale:

- ATI 3D RAGE PRO
- Matrox MGA Millenium II
- NeoMagic MagicGraph 128
- Number Nine Imagine 128, 9FX Vision 330 și Revolution 3D
- Compaq QVision 2000
- cipsetul Weitek P9100 (folosit la plăci video precum Diamond Viper Pro)
- cipseturile video Chips & Technologies 82C45x, 82C48x și F655xx

Multe sisteme Pentium PC sunt comercializate având incluse aceste plăci video de înaltă performanță. Deși probabil că în viitor se va include suport pentru cel puțin câteva dintre aceste plăci în sistemul XFree86, în prezent aceste produse comerciale vă permit să exploatați aceste plăci video sub X și Linux.

Serverele X comerciale asigură un suport superior pentru modurile cu 8, 16 și 32 bpp decât serverele XFree86. Aceste moduri se referă, în esență, la numărul de culori pe care îl puteți afișa simultan pe monitor. Modul 24 bpp permite un bait pentru fiecare din componentele unei culori RGB. Modul 32 bpp este – în esență – modul 24 bpp, dar valoarea RGB a unui pixel este stocată într-un cuvânt pe 32 de biți și 1 bait irosit (dar transferul de date pe 32 de biți este mai eficient decât transferul pe 24 de biți). Termenul 24 bpp este folosit pentru a specifica situația în care valorile RGB pe 24 de biți sunt stocate fără a irosi spațiu pe disc. În consecință, modul pe 24 de biți este deseori desemnat sub numele de mod 24 bpp packed-pixel („pixel împachetat“).

Rezumat

Plăcile video și monitoarele nu contează prea mult dacă folosiți Linux numai în mod text. Dacă doriți să folosiți sistemul X Window System XFree86, trebuie să acordați atenție plăcii video și monitorului. Acest capitol descrie modul de configurare a sistemului XFree86 și tipurile de plăci video și monitoare suportate.

Din lectura acestui capitol ați învățat următoarele:

- Într-un PC, placa video memorează matricea de pixeli care constituie imaginea pe care o vedeți pe ecran. Placa video convertește valorile de pixel în semnale analogice care direcționează tunurile electronice de roșu (R), verde (G) și albastru (B) asupra monitorului. Aceste fascicule RGB, la rândul lor, desenează imaginea colorată pe ecranul de display acoperit cu fosfor. Combinația dintre placa video și monitor este importantă pentru X Window System, deoarece serverul X controlează placa video în mod direct și deoarece monitorul trebuie să fie capabil de a controla semnalele generate de placa video.
- XFree86 include numeroase servere X, fiecare conceput pentru o anumită categorie de plăci video, de la plăcile comune VGA la plăci cu accelerare ca ATI Mach64 sau

S3. Trebuie să selectați serverul X în funcție de tipul plăcii video din sistemul dumneavoastră, deci trebuie să cunoașteți tipul plăcii de care dispuneți înainte de a instala XFree86.

- Când serverul X rulează, acesta consultă un fișier de configurare denumit `XF86Config` (situat de obicei în directorul `/etc`) pentru a selecta un mod video adecvat și pentru a configura placa video și monitorul în vederea unei funcționări corecte. Calculul unor parametri adecvați ai modului video este complicat. Dacă cunoașteți specificațiile tehnice ale monitorului dumneavoastră (adică frecvența de sincronizare pe orizontală și rata de reîmprospătare), nu trebuie să calculați aceste valori caracteristice modului video.
- Plăcile video se încadrează în două mari categorii. Toate plăcile Super VGA sunt controlate de serverul `XF86_SVGA`, iar plăcile video accelerate dispun de servere X adaptate, precum `XF86_S3` și `XF86_Mach64`.

Capitolul 11

Unități de disc

În acest capitol:

- ▶ Veți examina diferitele tipuri de controllere de disc suportate de Linux
- ▶ Veți înțelege conceptele unui hard-disc: cilindri, capete și sectoare (CHS)
- ▶ Veți înțelege operațiile efectuate cu unități de disc: partiționare și boot cu LILO
- ▶ Veți înțelege limita BIOS de 1024 de cilindri
- ▶ Veți examina tipurile de discuri SCSI suportate de Linux
- ▶ Veți cunoaște modul de depanare a unităților SCSI
- ▶ Veți folosi unități Iomega Zip în Linux
- ▶ Veți examina defecte cunoscute ale controllerelor de disc EIDE pentru magistrale PCI

Când vorbim despre unități de disc și Linux, important este faptul dacă Linux suportă sau nu *controllerul de disc*, adică placa prin care unitatea de disc este conectată la placa de bază a calculatorului dumneavoastră. Linux include suport pentru majoritatea controllerelor de disc. Acest capitol vă prezintă informații referitoare la modul de instalare și rulare a sistemului Linux pe un calculator dotat cu unul sau mai multe asemenea echipamente.

Tipuri de controllere de disc

Controllerul de disc este placa adaptor care se comportă ca un intermediar între placa de bază a calculatorului dumneavoastră și unul sau mai multe unități de hard-disc. De regulă, la un singur controller de disc se pot conecta maximum două unități de disc și două unități de dischetă. Controllerul Small Computer System Interface (SCSI) este o excepție de la regulă, deoarece permite conectarea a maximum șapte dispozitive SCSI (adică orice echipament care dispune de unități-SCSI, cum ar fi o unitate de disc, o unitate CD-ROM, o unitate de bandă sau un scanner).



Observație

De-a lungul anilor, au apărut mai multe tipuri de controllere de hard-disc pentru calculatoarele personale:

- controllerele de disc *ST506*, care au apărut la început în sistemele IBM XT și AT, au devenit opțiunea universală a industriei de PC-uri aflată la începuturi (să nu uităm că sistemul IBM PC-AT a apărut în 1984). Multe calculatoare dispun de controllere de disc compatibile ST506. Unitatea de disc Seagate ST506 a fost unitatea de disc originală a calculatoarelor, iar placa controller a acestora a fost Western Digital WD1003. Astfel, aceste controllere sunt deseori denumite *compatibile WD1003*. Unitățile originale ST506 foloseau o metodă de înregistrare cunoscută sub numele de modulare modificată în frecvență (Modified Frequency Modulation – MFM). Multe controllere de disc ST506 mai includ suport pentru unități care folosesc o altă tehnică de înregistrare a datelor, cunoscută sub numele de Run Length Limited (RLL).

- interfața *Integrated Drive Electronics (IDE)* emulează interfața ST506. Cu toate acestea, unitățile IDE conțin circuitele de controller încorporate chiar în unitate. Placa de bază conține, de regulă, o interfață IDE pentru conectarea unității de disc la placa de bază. În momentul de față, unitățile IDE sunt folosite pe scară largă în PC-uri. În prezent, termenul *AT Attachment (ATA)* este folosit pentru a desemna IDE. Interfața IDE originală putea suporta maximum două unități și limita dimensiunea maximă a discului la 500 MB. Calculatoarele moderne, cu unități de disc de mare capacitate, folosesc interfața Enhanced IDE, descrisă în continuare.

- Interfața *Enhanced IDE (EIDE)* suportă maximum patru periferice IDE interne (care pot fi unități de hard-disc, dar și unități CD-ROM), unități de disc de capacitate superioară și viteze superioare de transfer de date. Interfețele EIDE obișnuite sunt alcătuite din două interfețe IDE: primară și secundară, fiecare fiind capabilă de a suporta maximum două unități. Interfețele EIDE sunt frecvent folosite, datorită costului scăzut al acestora. Multe calculatoare folosesc interfața EIDE pentru a conecta atât unitatea de hard-disc, cât și unitatea CD-ROM, la placa de bază a calculatorului.

- controllerele *Enhanced Small Device Interface (ESDI)* emulează interfața ST506, dar asigură rate superioare de transfer al datelor.

- controllerele *Small Computer System Interface (SCSI)* asigură o magistrală separată, la care se pot conecta maximum șapte periferice SCSI, între care se numără unități de hard-disc, unități CD-ROM, unități de bandă și scannere care includ suport pentru SCSI. Puteți lega mai multe periferice SCSI la calculator prin conectarea în cascadă a acestora folosind un cablu SCSI. Toate stațiile de lucru UNIX (cum sunt cele produse de Hewlett-Packard, IBM și Sun Microsystems) folosesc SCSI. În ultima vreme, SCSI a devenit tot mai popular pe calculatoarele personale. Unicul dezavantaj este acela că un controller SCSI este mai costisitor în comparație cu controllerele EIDE.



Observație



Referință încrucișată

Linux include suport pentru toate aceste controllere de disc frecvent folosite. Chiar dacă în lista precedentă apar numeroase tipuri de controllere de disc, în esență există două tipuri de asemenea echipamente: IDE (unde sunt incluse și toate interfețele compatibile ST506) și SCSI.

Majoritatea sistemelor Pentium folosesc o placă de bază care include suport pentru magistrala de interconectare a componentelor periferice (Peripheral Component Interconnect – PCI). Sistemele PCI necesită plăci adaptor PCI care se conectează prin sloturi de magistrală PCI și o interfață IDE care se conectează la magistrala PCI. Linux include suport pentru magistrala PCI. Inițial, au apărut probleme legate de echipamentele cu interfață PCI, dar acum Linux poate rezolva unele dintre acestea. Se pare că magistrala PCI va deveni opțiunea dominantă pe piața calculatoarelor personale, înlocuind astfel vechea și demodată magistrală ISA. (În capitolul 9 am discutat despre diferite tipuri de magistrale.)

Noțiuni legate de unitatea de disc

Când citiți despre unități de disc, veți descoperi o terminologie și concepte caracteristice universului unic al hard-discurilor. În această secțiune vom explica unele dintre aceste concepte.

Cilindri, capete și sectoare

Organizarea fizică a discului este exprimată folosind termeni precum cilindri, capete și sectoare. Un hard-disc conține mai multe platouri din material magnetic. În termeni fizici, noțiunile de *cilindru*, *cap* și *sector* reprezintă următoarele:

- Un *cilindru* este unul dintre inelele concentrice situate pe o parte a platoului unui disc. Fiecare parte a platoului este divizată în cilindri.
- Numărul total de *capete* este numărul fețelor tuturor platourilor magnetice.
- Un *sector* este o zonă în formă de sector circular al platoului. Fiecare cilindru este divizat în sectoare.

Orice locație pe disc poate fi exprimată folosind succesiunea cilindru-cap-sector. Identificarea unei locații de disc folosind cilindrul (C), capul (head-H) și sectorul (S) este cunoscută sub numele de *adresare CHS*.



Geometria fizică a unui hard-disc este exprimată, de regulă, sub formă de cilindri, capete și sectoare. Un disc poate avea o geometrie de 528/32/63, ceea ce înseamnă că discul are 528 de cilindri, 32 de capete și 63 de sectoare. De regulă, fiecare sector poate stoca 512 baiți (sau jumătate de kilobait) de date. Astfel, capacitatea discului este $(528 \times 32 \times 63) / 2$ kilobaiți, sau 532.224 kilobaiți.



Observație Controllerele de hard-disc ale calculatorului includ o memorie read-only (ROM) BIOS (Basic Input Output System) de pe controller. Prin convenție (și compatibilitate cu arhitectura originală IBM PC), BIOS folosește adresarea CHS pentru accesarea discului. Cu toate acestea, BIOS-ul de disc folosește o valoare pe 10 biți ca adresă a cilindrului. Deoarece 10 biți pot memora numere cuprinse între 0 și 1023, BIOS-ul poate adresa maximum 1024 de cilindri.

Multe discuri de mari dimensiuni pot avea mai mult de 1024 de cilindri. Pentru a respecta limita de 1024 de cilindri, controllerele de disc trebuie să recurgă la unele trucuri pentru a manipula discuri cu un număr superior de cilindri. Ulterior, în secțiunea „Discuri cu mai mult de 1024 de cilindri” veți afla mai multe despre problemele discurilor cu mai mult de 1024 de cilindri, probleme care trebuie rezolvate.

Master Boot Record (MBR)

Primul sector al unui hard-disc (cilindrul 0, capul 0, sectorul 1) se numește *înregistrare master de boot* sau *Master Boot Record (MBR)*. Această suprafață de stocare de 512 baiți conține informații importante despre disc, cum ar fi tabela de partiție și un volum redus de cod, încărcat și rulat de BIOS-ul calculatorului la rularea secvenței de boot a sistemului.

Micul program situat în MBR citește tabela de partiție, determină partiția activă (acesta este numai unul dintre atributele unei partiții), citește primul sector al partiției active (*sectorul de boot*) și rulează orice program rezident în sectorul respectiv. Programul situat în sectorul de boot al unei partiții încarcă, de obicei, sistemul de operare instalat pe partiția respectivă.

Când instalați Linux Loader (LILO) pe hard-disc, programul LILO va fi rezident în MBR-ul discului sau în sectorul de boot al partiției unde este localizat directorul rădăcină Linux (/).

Partiții

Partițiile reprezintă o modalitate de a împărți un hard-disc și de a trata fiecare componentă separat. Prin împărțirea hard-discului unui calculator în partiții, puteți instala diverse sisteme de operare în diferite partiții. Chiar dacă folosiți întregul disc pentru Linux, aveți nevoie cel puțin de o partiție pe care Linux să o folosească drept *spațiu de schimb* (o extensie a memoriei), deci puteți dispune de o memorie virtuală cantitativ superioară memoriei fizice din sistem.



Master Boot Record conține tabela de partiții, care începe de la baitul 446 (0x1be, ceea ce înseamnă 1be în hexazecimal). Tabela de partiții poate avea maximum patru intrări de câte 16 baiți. Fiecare valoare pe 16 baiți definește o partiție. Fiecare partiție este exprimată printr-un număr de cilindru de început și de sfârșit. O intrare mai include un tip prin care este identificat sistemul de operare care a creat partiția. Conceptul de partiții este o convenție respectată de toate sistemele de operare utilizabile pe PC-uri, de la MS-DOS la Linux.

În MS-DOS se folosește programul FDISK pentru a manipula partițiile. Linux include un program cu același nume (fdisk – scris cu litere mici) pentru modificarea partițiilor de disc.

Nume de periferic atribuite discurilor în Linux

În Linux, fiecare dispozitiv (periferic) este reprezentat printr-un fișier de dispozitiv în directorul /dev. Numele de periferic aferent discului depinde de tipul de controller folosit de acesta. Pentru unitățile de disc IDE și EIDE, numele de dispozitiv este /dev/hda pentru primul disc, /dev/hdb pentru al doilea etc.

În cazul interfețelor EIDE, dacă dispuneți de un hard-disc pe interfața primară și o unitate CD-ROM pe interfața secundară, numele dispozitivelor sunt /dev/hda pentru unitatea de disc și /dev/hdb pentru unitatea CD-ROM.

Driverile de disc Linux tratează fiecare partiție ca un dispozitiv separat. Prima partiție a primului disc IDE este /dev/hda1, a doua partiție este /dev/hda2, a treia este /dev/hda3 și așa mai departe. Similar, numele de dispozitiv pentru partițiile din a doua unitate IDE sunt /dev/hdb1, /dev/hdb2 etc.

Dispozitivele tip disc SCSI sunt denumite /dev/sda, /dev/sdb etc. Dacă un dispozitiv SCSI este un hard-disc, partițiile sale sunt denumite prin atașarea numărului partiției la numele dispozitivului. Astfel, partițiile primei unități de disc SCSI se numesc /dev/sda1, /dev/sda2, /dev/sda3 etc.

Dischete în Linux

În capitolul 7 am descris numeroase modalități de accesare a dischetelor MS-DOS sub Linux. Pentru a citi din sau a scrie pe o dischetă, puteți monta discheta și folosi comenzile Linux sau puteți utiliza programele *mt001s*. De asemenea, puteți crea un fișier Linux pe o dischetă. De fapt, veți găsi sistemele de fișiere Linux pe dischetele de boot și root folosite pentru instalarea sistemului Linux.

Formatarea și crearea unui sistem de fișiere Linux pe o dischetă sunt procese simple. Pentru a formata o dischetă de 3,5 inch de înaltă densitate situată în unitatea A, de exemplu, folosiți următoarea comandă:

```
fdformat /dev/fd0H1440
```

Pentru o dischetă de 5,25 inch de înaltă densitate, numele de dispozitiv devine /dev/fd0H1200. Pentru unitatea B, primul 0 din numele de dispozitiv devine 1.



După ce formatați discheta, folosiți următoarea comandă pentru a crea un sistem de fișiere Linux pe dischetă:

```
mke2fs -m 0 /dev/fd0H1440 1440
```

Opțiunea `-m` se folosește pentru a specifica procentul de blocuri care urmează a fi rezervat pentru superutilizator (root). Prin specificarea opțiunii `-m 0`, programul `mke2fs` nu rezervă spațiu pe dischetă pentru superutilizator. Dacă nu specificați în mod explicit această opțiune, programul va rezerva 5% din spațiul pe disc pentru superutilizator.

După ce ați creat sistemul de fișiere pe dischetă, puteți monta discheta la un *punct de montare* (un director gol) din sistemul de fișiere Linux. Exemplul următor prezintă modul de montare a unității de dischetă în directorul `/mnt`:

```
mount /dev/fd0H1440 /mnt
```

Acum puteți folosi comenzile Linux, precum `cp` sau `mv`, pentru a copia sau muta fișiere pe dischetă. Înainte de a extrage discheta din unitate, folosiți următoarea comandă pentru a demonta unitatea:

```
umount /dev/fd0H1440
```

Operații cu discul în Linux

Pentru a instala și utiliza Linux pe sistemul dumneavoastră, trebuie să efectuați anumite operații cu discul. În capitolul 1 sunt explicate câteva dintre operațiile pe disc executate la instalarea sistemului Linux. În următoarele secțiuni vom furniza unele informații suplimentare referitoare la aceste operații.

În calculatoarele noi, hard-discul este de regulă configurat ca o singură partiție imensă, iar sistemele DOS și Windows sunt deja instalate. (Dacă v-ați cumpărat recent un calculator, acesta probabil include Windows 95 sau Windows 98 preinstalat.) Pentru a instala Linux, trebuie să începeți prin a crea minimum două partiții pentru Linux: una pentru spațiul de schimb și alta pentru sistemul de fișiere Linux.

Dacă dispuneți de dischete originale pentru reinstalarea sistemului de operare curent (indiferent dacă este Windows 95 sau MS-DOS și Windows 3.1), dați-i drumul și repartitionați discul. Pentru aceasta, încărcați sistemul de operare folosind o dischetă de boot DOS și rulați versiunea MS-DOS a programului `FDISK` pentru a șterge partiția curentă și a crea partiții noi. Creați trei noi partiții: una pentru DOS și Windows, una pentru spațiul de schimb Linux și una pentru sistemul de fișiere Linux. Apoi, trebuie să reinstalați DOS și Windows în partiția aferentă și să instalați Linux pe cealaltă partiție.



Referință
încrucțată

Dacă nu doriți să aveți din nou probleme cu reinstalarea sistemelor DOS și Windows sau Windows 95, va trebui să modificați într-un fel partiția existentă. Programul `FIPS`, care poate separa o partiție DOS existentă în două partiții separate, vă permite să executați această operație. Capitolul 1 descrie modul de utilizare a `FIPS`.

Partiționarea cu programul fdisk

Partiționarea discului implică crearea mai multor dispozitive logice mai mici în cadrul unui singur hard-disc. Sub MS-DOS se folosește programul `FDISK` pentru a vizualiza și modifica tabela de partiție a unui disc. În Linux, programul de partiționare se numește `fdisk`.



Referință
încrucțată

În capitolul 1 se explică modul de utilizare a programului `FDISK` pentru Linux și DOS în timpul instalării sistemului Linux. Am făcut această mențiune pentru completitudine și pentru a vă reaminti de programul `fdisk`.

Chiar dacă v-ați partiționat deja discul, puteți întotdeauna să rulați `fdisk` numai pentru a vedea tabela curentă de partiții pe un hard-disc. Dacă dispuneți de o unitate de disc SCSI cu numele

/dev/sda, de exemplu, puteți examina tabela sa de partiții folosind comanda `fdisk`, așa cum se vede în continuare:

```
fdisk /dev/sda
Command (m for help): m
Command action
  a toggle a bootable flag
  b edit bsd disklabel
  c toggle the dos compatibility flag
  d delete a partition
  l list known partition types
  m print this menu
  n add a new partition
  p print the partition table
  q quit without saving changes
  t change a partition's system id
  u change display/entry units
  v verify the partition table
  w write table to disk and exit
  x extra functionality (experts only)
```

```
Command (m for help): p
Disk /dev/sda: 64 heads, 32 sectors, 500 cylinders
Units = cylinders of 2048 * 512 bytes
```

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/sda1		1	1	301	308208	6	DOS 16-bit >=32M
/dev/sda2		302	302	499	202752	83	Linux native

```
Command (m for help): q
```

Comanda `m` prezintă utilizatorului o listă cu comenzile de o literă acceptate de `fdisk`. Puteți vedea tabela curentă de partiții cu o comandă `p`. Discul SCSI din acest exemplu are două partiții: una pentru DOS și una pentru Linux.

Câmpul `Id` din tabela de partiții afișată de programul `fdisk` (atunci când tastați `p` la prompt-ul de comandă) este un număr prin care se notează un tip de partiție. Dacă doriți să consultați o listă cu toate tipurile de partiție cunoscute, tastați `1` (adică `L` mic) la prompt-ul programului `fdisk`. Tabelul 11.1 prezintă tipurile de partiții pe care Linux le înțelege.

Tabelul 11.1 Tipuri de partiție cunoscute în Linux

Tip	Descriere	Tip	Descriere	Tip	Descriere	Tip	Descriere
0	Empty (Vidă)	9	AIX bootable	75	PC/IX	b7	BSDI fs
1	DOS 12-bit FAT	a	OS/2 Boot Manager	80	Old MINIX	b8	BSDI swap
2	XENIX root	b	Win95 FAT32	81	Linux/MINIX	c7	Syrinx
3	XENIX usr	40	Venix 80286	82	Linux swap	db	CP/M
4	DOS 16-bit < 32M	51	Novell?	83	Linux native	e1	DOS access

(continuare)

Tabelul 11.1 Continuare

Tip	Descriere	Tip	Descriere	Tip	Descriere	Tip	Descriere
5	Extended	52	Microport	93	Amoeba	e3	DOS R/O
6	DOS 16/bit >= 32	63	GNU HURD	94	Amoeba BBT	f2	DOS secondary
7	OS/2 HPFS	64	Novell Netware	a5	BSD/386	ff	BBT
8	AIX	65	Novell Netware				

Încărcarea sistemului de operare de pe disc folosind LILO

Pentru a încărca Linux în mod automat de pe hard-disc, aveți nevoie de programul Linux Loader (LILO). Acesta este un program de încărcare a secvenței de boot; sistemul OS/2 are un program echivalent, denumit *Boot Manager*. Aceste programe sunt de regulă rezidente în înregistrarea master de boot (Master Boot Record) a unui disc și sunt primele care se încarcă. Programul de încărcare a secvenței de boot vă solicită apoi numele unui sistem de operare de lansat (adică numele unei partiții de disc din care să încarce sistemul de operare). Lansarea unui sistem de operare implică, în esență, încărcarea în memorie a programului principal al sistemului respectiv și rularea acestuia. Pentru Linux, aceasta înseamnă încărcarea nucleului Linux în memorie și cedarea către nucleu a controlului asupra sistemului.



Referință încrucișată

LILO înseamnă mai multe decât Linux Loader; programul mai servește și ca gestionar de boot, capabil să încarce MS-DOS, OS/2 sau Windows 95. Capitolul 1 descrie modul de instalare LILO pe hard-disc în timpul procesului de instalare a sistemului Linux. În această secțiune vom descrie pe scurt procesul de instalare LILO în situația în care acesta se realizează în afara programului de instalare Linux, descris în capitolul 1.



Sugestie

Documentația LILO se află în directorul `/usr/doc/lilo-0.20` din sistemul dumneavoastră. Trebuie să consultați fișierul `README` din acest director pentru a afla ultimele noutăți referitoare la instalarea și configurarea LILO. În următoarea secțiune este prezentată numai o trecere în revistă a procesului.

Instalarea LILO

De regulă, LILO se instalează ca o componentă a procesului de instalare a sistemului Linux, descris în capitolul 1. Singura situație când trebuie să repetați configurarea și instalarea LILO o constituie actualizarea nucleului sau adăugarea unui nou sistem de operare pe care doriți să-l încărcați prin intermediul LILO.

Instalarea LILO implică două etape fundamentale:

1. Pregătirea fișierului de configurare LILO, care conține informații necesare pentru instalarea LILO. Fișierul prestabilit de configurare LILO este `/etc/lilo.conf`.
2. Rularea programului `sbin/lilo` (desemnat sub numele *map installer* în fișierul LILO `README`) în vederea actualizării sectorului de boot și crearea fișierului `/boot/map`, care conține informații utilizate de LILO în timpul procesului de boot.



Referință încrucișată

Pentru a pregăti fișierul de configurare LILO, trebuie să intrați în sistem ca root și să editați fișierul `lilo.conf`. În capitolul 2 este prezentată o scurtă descriere a modului de editare a fișierului respectiv. În esență, se adaugă sau se modifică informații cu privire la sistemele de operare ce urmează a fi încărcate cu Linux și la numele partițiilor de disc unde rezidă sistemele de operare respective.

Când fișierul de configurare LILO (`lilo.conf`) este gata, puteți instala LILO cu următoarea comandă:

```
/sbin/lilo
```

Programul `lilo` caută fișierul `lilo.conf` din directorul `/etc`, interpretează conținutul acestuia, pregătește fișierele de boot și map necesare și inițializează înregistrarea de boot a dispozitivului specificat (prin linia `boot`) în fișierul de configurare. Dacă dispozitivul de boot este specificat drept o partiție de hard-disc, de exemplu, LILO configurează sectorul de boot al partiției de disc respective.



Sugestie

Fișierele din directorul `/usr/doc/lilo-0.20`, cu precădere `/usr/doc/lilo-0.20/README`, conțin ultimele informații despre LILO. Este necesar să consultați acest fișier dacă dispuneți de un aranjament unic de partiții de hard-disc și doriți să aflați dacă puteți folosi LILO pentru a încărca Linux.

Utilizarea promptului de boot LILO

La lansare, LILO afișează prompt-ul boot : și așteaptă introducerea numelui unei imagini de boot pentru încărcare. Dacă nu răspundeți într-un interval de timp specificat (valoare stocată în fișierul `/etc/lilo.conf`), LILO încarcă imaginea de boot prestabilită (adică prima imagine din fișierul `lilo.conf`).



* Notă

Când folosiți LILO pentru a încărca sistemul Linux, puteți specifica numele imaginii de boot Linux, urmat de una sau mai multe opțiuni. LILO transferă aceste opțiuni din linia de comandă direct către nucleul Linux. Când citiți descrierile diferitelor tipuri de controllere de disc, veți descoperi informații despre opțiunile în linia de comandă de boot pe care le puteți folosi pentru a specifica diferiți parametri ai unui periferic. Între acești parametri se numără adresa portului I/O, IRQ-ul și DMA-ul unui periferic. Rețineți că aceste opțiuni din linia de comandă sesizează diferența între majuscule și literele mici.

Eliminarea LILO

Dacă instalați LILO în MBR și doriți să-l eliminați ulterior, trebuie să executați boot de pe o dischetă de boot și să rulați comanda `FDISK/MBR`, comandă ce readuce MBR la formatul MS-DOS. O altă modalitate de dezinstalare LILO este comanda `/sbin/lilo -u`.



Atenție!

Această comandă se folosește numai dacă ați instalat LILO în MBR. La opțiunea `-u`, LILO copiază un fișier denumit `/boot/boot.nnnn` (`nnnn` – numărul de periferic) în MBR. Dacă un fișier de boot vechi a rămas în directorul `/boot`, LILO poate copia fișierul respectiv în MBR. În cazul unui MBR defect, executați boot de pe o dischetă DOS și folosiți comanda `FDISK/MBR`.

Crearea spațiului de schimb

Spațiul de schimb este o partiție de disc folosită de Linux ca extensie de memorie, pentru stocarea unor date rezidente în memorie care nu sunt necesare imediat. Pentru crearea acestui spațiu, creați o partiție folosind `fdisk` (partiție setată ca Linux swap).

Dacă trebuie să setați spațiul de schimb în afara programului de instalare, urmați procedura din Capitolul 1. Veți utiliza comanda `mkswap` pentru inițializarea partiției de schimb, precedată de

programul `fdisk` pentru aflarea dimensiunii partiției. Apoi, folosiți comanda `swapon` pentru a activa această caracteristică. Pentru a verifica dacă Linux folosește spațiul de schimb la fiecare boot, aveți nevoie de următoarea linie în fișierul `/etc/fstab`:

```
/dev/hda2    swap    swap    defaults    0 0
```



Dacă în `/etc/fstab` se află numele unei partiții pe care nu s-a rulat `mkswap`, veți primi următorul mesaj de eroare:

Secret

```
Unable to find swap - space signature
```

Crearea sistemelor de fișiere

Pentru a folosi o partiție de disc în Linux, trebuie să creați un sistem de fișiere pe acea partiție. Pentru o partiție Linux, programul de instalare folosește comanda `mk2efs` pentru a crea un sistem de fișiere Linux.

Linux include suport pentru mai multe tipuri de sisteme de fișiere:

- Sistem de fișiere MS-DOS. Este bazat pe FAT și specificat de cuvântul cheie `msdos`.
- Sistem de fișiere Minix. Limitează numele de fișiere la 14-30 caractere. Identificat prin cuvântul cheie `minix`.
- Sistem de fișiere extins. Identificat prin cuvântul cheie `ext`. Nu se mai folosește.
- Al doilea sistem de fișiere extins. Cel mai important. Permite nume de fișier lungi. Cuvânt cheie: `ext2`.

Programul de instalare creează în mod automat un sistem `ext2` pe partiția Linux. Pentru a crea sistemul pe cale manuală, folosiți `fdisk` și apoi comanda

```
mk2efs - c/dev/hda3 405040,
```

care creează un sistem de fișiere `ext2`, ce conține 308040 de blocuri și detectează blocurile cu erori.

Probleme de disc specifice Linux

Pentru majoritatea unităților de disc, instalarea sistemului Linux se reduce la încărcarea unui nucleu Linux care suportă controllerul de hard-disc al sistemului, partiționarea discului sub Linux și încărcarea sistemului de operare și a programelor asociate pe hard-disc. De regulă, discul este luat în seamă numai la crearea partițiilor (așa cum s-a arătat în capitolul 1).

Cu toate acestea, când vreunul din aspectele legate de hard-disc devine neobișnuit, puteți avea probleme la instalarea sistemului Linux. În această secțiune sunt descrise unele din aceste probleme și sunt propuse soluții.

Windows 95 și LILO

Pe un PC care dispune de Linux și Windows 3.1 instalate în partiții separate, puteți folosi programul Linux Loader (LILO) pentru a încărca unul sau altul dintre sistemele de operare. Dacă treceți de la Windows 3.1 la Windows 95, veți descoperi că Windows 95 șterge LILO. Windows 95 suprascrie înregistrarea master de boot (Master Boot Record) cu propriul său program, iar LILO este rezident, de obicei, în MBR. În acest caz, nu trebuie decât să încărcați Linux folosind o dischetă de boot (pe care ați creat-o la instalarea sistemului Linux), după care rulați `/sbin/lilo` pentru a reinstala LILO.

Dacă ați achiziționat un calculator nou cu Windows 95 preinstalat, tot ce aveți de făcut este să instalați Linux prin parcurgerea etapelor descrise în capitolul 1 (începeți prin repartiționarea

hard-discului). Când ajungeți la etapa de instalare LILO, specificați sistemele de operare și partițiile de disc pe care LILO trebuie să le configureze pentru boot. Când specificați partițiile de boot, tratați partiția de disc Windows 95 ca pe o partiție DOS. După ce ați încheiat instalarea sistemului Linux și a programului LILO, trebuie să puteți încărca Windows 95 (presupunând că l-ați lăsat instalat) sau Linux prin introducerea numelui imaginii de boot corespunzătoare la promptul de boot LILO.

Discuri cu peste 1024 de cilindri

Discurile care dispun de peste 1024 de cilindri reprezentau o problemă pentru versiunile mai vechi de nuclee Linux. Cu toate acestea, versiunea existentă pe CD-ROM-ul anexat cărții este compatibilă cu discuri EIDE cu mai mult de 1024 de cilindri, deci puteți ignora în liniște paragrafele acestei secțiuni.



Vă puteți imagina un disc ca pe o colecție de sectoare, fiecare sector având câte 1024 baiți. Sectoarele sunt adresate în două moduri:

- adresă liniară de bloc (Linear Block Address - LBA), în care sectoarele sunt numerotate secvențial, începând de la zero.
- adresă cilindru-cap-sector, bazată pe construcția fizică a unui hard-disc, alcătuită din platouri magnetice care se rotesc la turații mari sub capetele de citire-scriere.

Controllerle de disc vechi pentru PC și sistemul BIOS (Basic Input/Output System) folosesc adresele CHS fizice pentru accesarea sectoarelor de pe disc. Acest aranjament a fost funcțional până la apariția pe piață a discurilor de mari dimensiuni. BIOS folosește o valoare pe 10 biți pentru numărul cilindrului, ceea ce înseamnă că poate accesa un maximum de 1024 de cilindri de pe hard-disc (deoarece o valoare pe 10 biți este cuprinsă între 0 și 1023). Din cauza acestei limitări, orice disc cu mai mult de 1024 de cilindri este desemnat sub numele de *disc mare* (*large disk*) în lumea calculatoarelor personale.



Majoritatea hard-discurilor curente au 16 capete, 63 de sectoare și un număr mare de cilindri. Cu limitarea de 1024 de cilindri, dimensiunea maximă a unui disc sub BIOS este $16 \times 63 \times 1024 = 1.032.192$ sectoare = 528.482.304 baiți (deoarece fiecare sector conține 512 baiți) = 504 MB (1 MB = 1.048.576 baiți). După cum știți, multe calculatoare de birou actuale dispun de discuri de 2 GB sau 4 GB. De regulă, aceste discuri dispun de o interfață de tip Enhanced IDE (EIDE).

Interfața EIDE manipulează discuri de mari dimensiuni prin ajustarea numărului de cilindri și de capete. Un truc folosit de BIOS-ul EIDE este de a reduce la jumătate numărul real de cilindri și de a dubla numărul de capete. Când BIOS procesează o cerere de acces la disc specificată de adresa CHS, acesta ajustează în mod automat numerele de cilindri și de capete pentru a accesa sectorul corect de pe disc. Această ajustare se numește *conversia adreselor* (*address translation*).

Linux preia geometria unui disc (număr de cilindri, capete și sectoare) din BIOS. Dacă BIOS execută conversia adreselor, valorile raportate nu se vor corela cu parametri fizici reali ai discului. Din păcate, Linux nu trece prin BIOS pentru a citi de pe sau a scrie pe disc. Când accesează controllerul de disc, Linux trebuie să furnizeze adresa CHS cu valorile fizice corecte. Cu alte cuvinte, Linux trebuie să execute conversia de adrese așa cum o execută BIOS-ul. Din fericire, nucleul Linux de pe discul anexat acestei cărți a rezolvat toate aceste detalii și manipulează corect discurile de mari dimensiuni.

Cu toate acestea, mai puteţi avea probleme cu discurile mari, şi aceasta nu deoarece Linux nu poate manipula discurile de mari dimensiuni, ci deoarece pe multe PC-uri mai vechi, de tip MS-DOS, multe discuri vechi de mari dimensiuni sunt gestionate de programe speciale, care pot intra în conflict cu Linux. Unele sisteme mari sunt dotate cu un program special, cunoscut sub numele de Disk Manager. Acesta se găseşte în MBR-ul discului şi execută nişte trucuri pentru a permite sistemului DOS să acceseze hard-discul.

Unica soluţie „curată” pentru asemenea probleme necesită execuţia unor copii de siguranţă ale fişierelor dumneavoastră DOS. Setăţi parametrii de disc din BIOS la valorile corecte şi apoi partiţionaţi discul sub Linux. Puteţi atribui prima partiţie pentru DOS (deoarece acesta este incapabil de a accesa mai mult de primii 1024 de cilindri), după care creaţi partiţiile necesare pentru Linux dincolo de partiţia DOS.

Deoarece programul Linux Loader (LILO) se bazează pe BIOS, care nu poate accesa mai mult de 1024 de cilindri, încercaţi să creaţi partiţia de boot Linux în cadrul primilor 1024 de cilindri.

Probleme EIDE pe sisteme PCI

Generaţia actuală de sisteme Pentium foloseşte magistrala PCI. O dată cu aceasta au apărut şi noi tipuri de interfeţe pentru discuri. O interfaţă frecvent folosită este controllerul EIDE pentru PCI de conectare a perifericelor EIDE, cum sunt hard-discurile şi unităţile CD-ROM, la placa de bază PCI.



Secret

Când magistrala PCI a devenit populară în 1995, unii utilizatori au semnalat o corupere a datelor din discurile EIDE dotate cu anumite controllere EIDE PCI. Concret, sistemele afectate dispun de plăci de bază PCI cu controllere PCI EIDE care folosesc unul dintre următoarele cipuri:

- RZ 1000
- CMD 640

Pentru a afla care este tipul de controller EIDE PCI folosit de sistemul dumneavoastră, utilizaţi comanda `cat /proc/pci` (cu condiţia, evident, ca sistemul dumneavoastră să folosească o magistrală PCI).

Versiunea Linux de pe CD-ROM-ul asociat acestei cărţi detectează şi rezolvă aceste probleme în mod automat. Unica reminiscenţă a problemei este că, atunci când tastaţi comanda `cat /pci/proc`, listingul arată că controllerul CMD 640 este defect. De exemplu, iată primele câteva linii de ieşire pe unul din calculatoarele mele, dotate cu interfaţa CMD 640:

```
cat /proc/pci
PCI devices found:
  Bus 0, device 13, function 0:
    IDE interface: CMD 640 (buggy) (rev2).
    Medium devsel. IRQ 14.
(celelalte linii au fost sterse)
```

Observaţi că interfaţa CMD 640 este semnalată ca fiind defectă.

Mesaje de eroare privind inoduri şi blocuri

Dacă primiţi mesaje de eroare privind inoduri sau blocuri defecte în timpul lansării sistemului, este posibil ca sistemul să nu fi fost corect închis. Înainte de a întrerupe alimentarea unui sistem Linux, întotdeauna trebuie să intraţi în sistem ca root şi să folosiţi comanda de închidere pentru a opri sistemul, aşa cum se vede în continuare:

```
/sbin/shutdown -h now
```

După această comandă, trebuie să aşteptaţi până la apariţia unui mesaj prin care se spune că sistemul a fost oprit. Numai după această comandă puteţi deconecta calculatorul de la reţea.

Datorită experienţei noastre cu apăsarea butonului la sisteme controlate de MS-DOS, majoritatea suntem tentaţi să apăsăm acelaşi buton pentru a închide sistemul (deşi, într-un fel, îmi amintesc întotdeauna să închid corect staţiile de lucru UNIX, cum sunt sistemele HP sau Sun).

Dacă sistemul nu este corect închis, sistemul de fişiere poate fi alterat. Când încărcaţi sistemul de operare data viitoare, acesta va rula un program de verificare a sistemului de fişiere (`fsck`) care poate remedia sistemul de fişiere şi va încărca sistemul de operare. În unele situaţii, totuşi, `fsck` va fi incapabil de a remedia pagubele provocate sistemului de fişiere. Nu aveţi altă soluţie decât să reiniţializaţi sistemul de fişiere folosind comanda `mke2fs`.

Controllere de disc SCSI şi Linux

În continuare, acest capitol va prezenta modul de utilizare a controllerelelor SCSI sub Linux.

SCSI (pronunţat *scazi*) este o interfaţă tot mai populară pentru conectarea a maximum şapte periferice diferite pe magistrala SCSI. Fiecare periferic şi controllerul SCSI dispun de un identificator SCSI unic (ID) cuprins între 0 şi 7. Controllerul primeşte, de obicei, identificatorul 7, iar celelalte periferice folosesc numere cuprinse între 0 şi 6 (aceasta înseamnă că puteţi conecta maximum şapte periferice la un controller SCSI). De regulă, un hard-disc SCSI are identificatorul setat la 0.

Linux include suport pentru următoarele controllere SCSI:

- Adaptec AHA-1510/152x (ISA)
- Adaptec AHA-154x (ISA)
- Adaptec AHA-174x (EISA)
- Adaptec AHA-274x (EISA)/284x (VLB)
- Adaptec AHA 2920
- Adaptec AHA-2940/3940 (PCI)
- Adaptec AVA-1505/1515 (ISA) (compatibil Adaptec 152x)
- Always IN2000
- AMI Fast Disk VLB/EISA (compatibil BusLogic)
- BusLogic (ISA/EISA/VLB/PCI) (toate modelele)
- DPT PM2001 şi PM2012A (EATA-PIO)
- DPT Smartcache (EATA/DMA) (ISA/EISA/PCI)
- DTC 329x (EISA) (compatibil Adaptec 154x)
- Future Domain TMC-16-x0 şi TMC-3260 (PCI)
- Future Domain TMC-8xx şi TMC-950
- ICP-Vortex (PCI/EISA)
- Media Vision Pro Audio Spectrum 16 (ISA)
- Media Vision Premium 3D
- NCR 53c7x0 şi 53c8x0 (PCI)
- Plăcile generice NCR 5380
- Qlogic FAS408

Quantum ISA-200S și ISA-200MG

Seagate ST-01/ST-02 (ISA)

Sound Blaster 16 SCSI (compatibil Adaptec 152x) (ISA)

Tekram DC-390, DC-390W/U/F

Trantor T128/T128F/T228 (ISA)

Trantor T130B (compatibil NCR 5380)

UltraStor 14F (ISA), 24F (EISA) și 34F (VLB)

Western Digital WD7000 SCSI

Linux nu include suport, de obicei, pentru adaptoare SCSI de porturi paralele și plăci DTC incompatibile Adaptec (cum sunt 327x și 328x).



Referință
încrucișată

Când configurați nucleul Linux pentru suportul SCSI, programul de configurare pune un număr de întrebări referitoare la plăcile controller SCSI. Consultați capitolul 2 pentru a vedea un listing de întrebări despre suportul SCSI. Trebuie să răspundeți afirmativ (tastați **y**) numai la întrebarea care face referire la producătorul și modelul plăcii SCSI.

Probleme de cablu și terminatori

Magistrala SCSI necesită terminatori la ambele capete pentru a putea funcționa corect. Un *terminator* este un set de rezistori care indică capătul magistralei SCSI. Un capăt este placa controller însăși, care conține și un terminator. Fiecare periferic SCSI dispune de doi conectori SCSI, deci puteți conecta în lanț (unul după celălalt) numeroase periferice SCSI. Ultimul conector din lanț trebuie să conțină și terminatorul.



Secret

Unele controllere SCSI – precum Adaptec AHA 154xC, 154xCF și 274x (x este orice cifră) sunt sensibile la tipul de cablu și terminator folosit. Dacă cablurile conțin imperfecțiuni sau terminatoarele nu sunt corect folosite, aceste plăci SCSI pot avea căderi intermitente sau pot deveni complet nefuncționale.

Pentru a evita problemele cu plăcile SCSI extrem de sensibile, folosiți cabluri produse de o firmă cunoscută și utilizați cabluri de la aceeași firmă pentru a conecta toate perifericele SCSI. Cablurile trebuie să fie compliante SCSI-2 și trebuie să aibă o impedanță de 132 de ohmi (o caracteristică a cablului; tot ce aveți de făcut este să verificați dacă valoarea specificată este de 132 de ohmi).

Adaptec AHA151x și Sound Blaster 16 SCSI

Aceste plăci SCSI pe magistrală ISA includ toate controllerele SCSI bazate pe cipsetul AIC 6260 sau 6360. Parametrii hardware obișnuiți pentru aceste controllere includ următoarele:

adrese BIOS: 0xd8000, 0xdc000, 0xd0000, 0xd4000, 0xc8000, 0xcc000, 0xc0000 și 0xe4000

porturi I/O: 0x140 și 0x340

IRQ: 9, 10, 11 și 12

canale DMA: nefolosite



Secret

Autoprobe funcționează cu plăci care au BIOS instalat. Pentru alte plăci, precum Adaptec 1510 și Sound Blaster 16 SCSI, folosiți opțiunea de boot în următorul format:

aha152x=IOPORT, IRQ, SCSI-ID, RECONNECT

Toate argumentele din membrul drept sunt numere. *IOPORT* este adresa de I/O, iar dacă *RECONNECT* este diferit de zero, driverul are permisiunea de a deconecta și apoi reconecta perifericul. De regulă, *SCSI-ID* este 7, iar *RECONNECT* are valoarea 1.

De regulă, *SCSI-ID* este 7, iar *RECONNECT* este diferit de zero. Dacă o placă Adaptec AHA1510 are adresa I/O 0x340 și IRQ 11, opțiunea de boot este următoarea:

aha152x=0x340, 11, 7, 1

Adaptec AHA154x, AMI FastDisk VLB, BusLogic și DTC 329x

Parametrii hardware specifici acestor controllere SCSI pe magistrală ISA sunt următorii:

porturi I/O: 0x330 și 0x334

IRQ: 9, 10, 11, 12, 14 și 15

canale DMA: 5, 6 și 7

Autoprobe funcționează cu aceste controllere; nu este necesar BIOS-ul pe controller. Controllerele SCSI BusLogic sunt compatibile software cu Adaptec 1542. Există versiuni BusLogic pentru ISA, VLB și EISA.



Secret

Plăcile controller Adaptec AHA 154xC și AHA154xCF generează deseori erori neașteptate; aceste controllere sunt sensibile la detalii legate de cablu și terminatori.

Dacă apar erori de timeout infinit la plăcile Adaptec AHA154xC și AHA154xCF, trebuie să rulați programul de configurare Adaptec (prin apăsarea unei anumite taste în timpul inițializării) și să activați negocierea sincronă.

Adaptec AHA 174x

Acest controller SCSI pe magistrală EISA nu mai este produs de Adaptec. Sistemele mai vechi pe magistrală EISA pot dispune de această placă. Parametrii hardware ai plăcii sunt următorii:

sloturi de magistrală: 1-8

porturi I/O: magistrala EISA nu necesită porturi I/O preatribuite

IRQ: 9, 10, 11, 12, 14 și 15

canale DMA: magistrala EISA nu necesită canale DMA preatribuite

Driverul Linux poate detecta placa în mod automat, fără probleme. De asemenea, driverul se așteaptă ca placa să ruleze în mod enhanced, nu în modul standard AHA1542.

Adaptec AHA274x, AHA284x și AHA294x

Controllerul Adaptec AHA274x este o placă de magistrală EISA, AHA284x este o placă VLB, iar AHA294x este o placă nouă, pentru magistrală PCI. Driverul AHA274x include suport pentru toate trei plăcile. Trebuie activat BIOS-ul pe aceste plăci.



Pentru ca controllerul PCI să funcționeze, trebuie să răspundeți afirmativ la următoarea întrebare în timpul configurării nucleului:

PCI bios support (CONFIG_PCI) [Y/n/?] y

Allways IN2000

Această placă este un controller pe magistrală ISA, cu următorii parametri:

porturi I/O: 0x100, 0x110, 0x200 și 0x220

IRQ: 10, 11, 14 și 15

DMA: nefolosit

Driverul poate detecta placa în mod automat, fără a fi nevoie de BIOS.

EATA DPT Smartcache

Driverul Linux `eata_dma` include suport pentru toate controllerele SCSI care suportă protocolul EATA-DMA. Aceste controllere sunt: DPT PM2011, PM2012A, PM2012B, PM2021, PM2022, PM2024, PM2122, PM2124, PM2322, PM3021, PM3222 și PM3224.



Secret

Funcția autoprobe a driverului funcționează cu toate plăcile DPT suportate. O problemă comună, însă, este aceea că driverul IDE detectează interfața ST-506 a controllerului EATA. Dacă driverul IDE are o problemă cu parametrii detectați și eșuează, nu veți putea accesa echipamentul IDE. În acest caz, trebuie să modificați parametrii plăcii EATA, cum ar fi adresa I/O sau IRQ. În particular, nu folosiți IRQ 14 sau 15, care sunt IRQ-urile folosite de interfețele IDE primară și secundară.

Dacă dispuneți de un controller PCI ca DPT PM2024, PM2124 sau PM3224, nu uitați să activați PCI BIOS la configurarea nucleului.

Future Domain 16x0

Controllerul Future Domain 16x0 folosește unul din cipurile TMC-1800, TMC-18C30, TMC-8C50 sau TMC-36C70. Aceste plăci ISA folosesc, de regulă, următoarele configurații:

adrese BIOS: 0xc8000, 0xca000, 0xce000 și 0xde000

porturi I/O: 0x140, 0x150, 0x160 și 0x170

IRQ: 3, 5, 10, 11, 12, 14 și 15

DMA: nefolosit

Driverul poate proba și detecta echipamentul în mod automat, cu condiția ca controllerul să aibă BIOS instalat.

Cipul SCSI NCR53c8xx (PCI)

Prin *NCR53c8xx* se înțeleg cipurile NCR53c810, NCR53c815, NCR53c820 și NCR53c825, o serie de cipuri SCSI necostisitoare pentru plăci de bază PCI. Versiunea Linux de pe CD-ROM-ul asociat cărții include suport pentru seria NCR53c8xx. Driverul poate detecta dispozitivele SCSI în mod automat, cu condiția existenței PCI BIOS. De fapt, driverul are nevoie de BIOS deoarece folosește valori inițializate de BIOS în registrele cipului NCR53c8xx.



Secret

O problemă semnalată vizavi de seria NCR53c8xx este faptul că cipul este funcțional sub DOS, dar nu și sub Linux, deoarece depășește intervalul de timp la un test datorită unei întreruperi pierdute. O cauză obișnuită a acestei erori o constituie o neechivalență între setarea IRQ din hardware (setat de regulă folosind un jumper) și valoarea IRQ stocată în CMOS (programul de configurare a calculatorului, care poate fi rulat în timpul inițializării sistemului). Pentru a remedia problema, puteți verifica următoarele:

■ Setarea IRQ din hardware trebuie să corespundă celei existente în CMOS.

■ Dacă NCR53c8xx se găsește pe o placă ce conține jumperi pentru selectarea liniilor de întrerupere PCI (PCI are ca linii de întrerupere INTA, INTB, INTC și INTD), verificați ca numai INTA să fie în uz.

■ Dacă placa PCI are jumperi pentru selectarea întreruperilor sensibile la nivel (level-sensitive) sau a celor declanșate pe frontul semnalului (edge-triggered), verificați ca placa să folosească numai întreruperi sensibile la nivel.



Secret

O altă problemă semnalată este blocarea sistemului la utilizarea unui cipset video PCI de tip S3 928 sau Tseng ET4000/W32, datorită neajunsurilor cipseturilor video.



Secret

Pe un sistem care conține un cip SCSI NCR53c8xx, puteți întâlni următorul mesaj:

`scsi%d: IRQ0 not free, detaching`

Aceste mesaj arată că registrul de configurare PCI conține un zero. Cauza poate fi o nepotrivire între IRQ setat hardware și valoarea existentă în CMOS, dar și un BIOS cu erori.

Deoarece placa NCR53c8xx este un periferic PCI, trebuie să activați suportul PCI la regenerarea nucleului Linux.

Seagate ST0x și Future Domain TMC-8xx și TMC-9xx

Parametrii hardware ai acestor controllere SCSI pe magistrale ISA au următoarele valori:

adrese BIOS: 0xc8000, 0xca000, 0xcc000, 0xce000, 0xcd000 și 0xde000

IRQ: 3 și 5

canale DMA: nefolosite

Când încearcă să probeze controllerul în mod automat, driverul probează numai adresele BIOS; se presupune că IRQ este 5. De asemenea, autoprobația funcționează numai dacă este instalat un BIOS.

În timpul execuției secvenței de boot, puteți da una din următoarele comenzi pentru a forța detecția controllerului:

```
st0x=BIOS_ADDRESS, IRQ
tmc8xx=BIOS_ADDRESS, IRQ
```

`BIOS_ADDRESS` este adresa BIOS a plăcii, iar `IRQ` este canalul de solicitare a întreruperii.



Secret

Probleme frecvent apărute la utilizarea controllerelor SCSI ST01 sau ST02 sunt timeout-urile apărute când Linux accesează discul conectat la controller, deoarece setările prestabilite ale plăcilor dezactivează întreruperile. Trebuie să setați jumperele de pe plăci (W3 pe ST01 și JP3 pe ST02) pentru a reactiva întreruperile. De asemenea, trebuie să selectați IRQ 5.



Secret

Dacă aveți erori când încercați să rulați `fdisk` pe o unitate de disc conectată la unul din controllele Seagate sau Future Domain, trebuie să folosiți meniul funcțiilor suplimentare al programului pentru a specifica geometria discului (cilindri, capete și sectoare).

Pro Audio Spectrum PAS16 SCSI

Notăția PAS16 SCSI se referă la interfața SCSI de pe o placă de sunet Pro Audio Spectrum. În continuare sunt prezentați parametri de configurare hardware pentru placa PAS16 SCSI:

porturi I/O: 0x388, 0x384, 0x38x și 0x288

IRQ: 10, 12, 14 și 15 (trebuie să fie diferite de cele folosite pentru sunet)

DMA: nefolosite pentru porțiunea SCSI a plăcii



Funcția de autoproabă nu necesită BIOS. Puteți specifica o linie de comandă la prompt-ul de boot pentru a specifica parametrul plăcii PAS16 SCSI. Pentru o placă PAS16 SCSI la adresa I/O 0x388 și IRQ 10, de exemplu, folosiți următoarea linie de comandă:

`Pas16=0x388,10`

Trantor T128, T128F și T228

Aceste plăci SCSI pe magistrală ISA au următorii parametri de configurare:

adrese BIOS: 0xcc000, 0xc8000, 0xdc000 și 0xd8000

IRQ: pe toate plăcile, nici unul, 3, 5 și 7; pe T128F, 10, 12, 14 și 15

DMA: nefolosit



Driverul poate executa autoproabă dacă este instalat BIOS. Dacă unul dintre aceste controlle SCSI nu dispune de BIOS, sau dacă BIOS este dezactivat, puteți specifica controllerul prin intermediul unei linii de comandă, ca de exemplu:

`T128=BIOS_ADDRESS,IRQ`

`BIOS_ADDRESS` este adresa de bază (nu adresa I/O). Pentru un controller cu o adresă BIOS 0xcc000 și IRQ 5, de exemplu, linia de comandă este:

`T128=0xcc000,5`

Utilizați -1 pentru IRQ dacă controllerul nu dispune de un IRQ, respectiv -2 pentru a cere driverului să probeze IRQ.

Ultrastor 14f (ISA), 24f (EISA) și 34f (VLB)

Plăcile SCSI Ultrastor au următorii parametri de configurare:

porturi I/O: 0x130, 0x140, 0x210, 0x230, 0x240, 0x310, 0x330 și 0x340

IRQ: 10, 11, 14, 15

canale DMA: 5, 6 și 7

Funcția de autoproabă funcționează în toate cazurile, cu excepția cazului când adresa portului I/O este 0x310. Deoarece portul I/O 0x310 nu este suportat de codul de autoproabă, trebuie să selectați o altă adresă de port pentru controllerul Ultrastor.



Dacă dispuneți de o placă de sunet, adresa de port I/O 0x330 este folosită de perifericul MIDI. Folosiți o altă adresă I/O pentru placa Ultrastor dacă dispuneți de o placă de sunet. O opțiune recomandată pentru plăcile Ultrastor este 0x340.

Controllele Ultrastor includ suport pentru un mod de emulare WD1003, mod în care pot lucra cu unitățile de disc cu interfață ST-506. Dacă controllerul Ultrastor se află în mod WD1003, driverul Ultrastor SCSI va eșua, afișând următorul mesaj de eroare:

`hd.c: ST-506 interface disk with more than 16 heads detected, probably due to non-standard sector translation. Giving up. (disk %d: cyl=%d, sect=63, head=64)`

(Disc cu interfață ST-506 cu mai mult de 16 capete detectat, probabil datorită unei conversii non-standard de sectoare. Renunț.)

Această problemă poate fi remediată prin setarea controllerului Ultrastor la modul său nativ SCSI.

Western Digital 7000

Configurația hardware a acestui controller SCSI pe magistrală ISA sunt următoarele:

adresă BIOS: 0xce000

port I/O: 0x350

IRQ: 15

canal DMA: 6

Driverul poate proba și detecta controllerul SCSI automat, cu condiția ca BIOS să fie instalat.

Este posibil ca unele versiuni ale controllerului Western Digital 7000 să nu funcționeze cu driverul. Se pare că Revision 5 și controllele ulterioare funcționează corect. De asemenea, la controllele funcționale, cipul SCSI de pe placă trebuie să aibă sufixul A.

Unitatea Iomega Zip (SCSI)

Unitatea Iomega Zip este o unitate de disc necostisitoare, amovibilă, care vă permite să folosiți discuri similare cu dischetele, capabile de a stoca fiecare peste 100 de milioane de caractere. Ținând cont că 1 megabait = $1024 \times 1024 = 1.048.576$ baiți, atunci 100 milioane de caractere reprezintă aproximativ 95 MB. Spre deosebire de dischete, discurile Zip nu dispun de un element de protecție la scriere hardware, ceea ce înseamnă că trebuie să fiți atenți la inițializarea unui disc sau la ștergerea fișierelor.

Unitatea Zip se prezintă în trei versiuni: interfața SCSI pentru Macintosh sau MS-DOS și o versiune pentru porturi paralele. Nucleul Linux curent suportă versiunea pentru porturi paralele a unității Zip.

Atât versiunea DOS, cât și cea Macintosh a unității SCSI Iomega Zip trebuie să fie funcționale sub Linux. Puteți folosi o placă SCSI suportată de Linux (de regulă, o placă simplă compatibilă Adaptec AHA152x) pentru a conecta unitatea Zip la PC-ul dumneavoastră. Unitatea Zip dispune de comutatori pentru activarea terminării (dacă este ultimul periferic din lanț) și pentru selecția identificatorului SCSI (5 sau 6).

De asemenea, puteți conecta unitatea Zip la o placă de interfață Zip Zoom comercializată separat, în speță o placă SCSI cu performanțe reduse compatibilă Adaptec AHA152x. Dacă dispuneți de un nucleu care suportă AHA152x și folosiți LILO, puteți folosi unitatea Zip cu următoarea linie adăugată la fișierul `/etc/lilo.conf`:

`Append="aha152x=0x340,11,7,1"`



Sugestie

Pentru a pregăti discul Zip în vederea utilizării sub Linux, trebuie să îl încercați sub DOS. Rulați programul /scsi/install de pe discul Iomega Tools. După aceea, discul Zip trebuie să funcționeze sub Linux.

Întrați în sistem ca root și rulați /sbin/fdisk pe perifericul SCSI care reprezintă unitatea Zip. Dacă unitatea Zip este unicul periferic SCSI, va avea numele /dev/sda, deci tasteți următoarele:

```
/sbin/fdisk /dev/sda
```

Verificați partiția curentă și setați tipul acesteia la partiție nativă Linux. Apoi creați un sistem de fișiere ext2 folosind următoarea comandă:

```
/sbin/mke2fs /dev/sda1
```

Această comandă presupune că folosiți prima partiție a discului Zip. Creați un subdirector zip în /mnt cu comanda `mkdir /mnt/zip`. Apoi puteți monta discul Zip la un punct de montare adecvat și puteți folosi discul după cum urmează:

```
mount -text2 /dev/sda1 /mnt/zip
```

Remediarea problemelor cu perifericele SCSI

Majoritatea problemelor SCSI se datorează cablurilor cu defecte sau unei terminări necorespunzătoare. Trebuie să verificați cablurile și terminatorul înainte de a încerca orice altă soluție. În secțiunile următoare sunt prezentate alte probleme SCSI comune și câteva remedii propuse:

Probleme la boot cu LILO

Când executați boot de pe un hard-disc SCSI, LILO poate bloca sistemul după afișarea literelor LI. Această problemă survine dacă BIOS-ul controllerului SCSI și driverul SCSI pentru Linux interpretează geometria discului în mod diferit.



* Sfaturi

Pentru a remedia această problemă, adăugați cuvântul cheie `linear` pe o singură linie în fișierul /etc/lilo.conf. Acest cuvânt cheie determină LILO să folosească adresele liniare de bloc (Linear Block Addresses - LBA) în locul adreselor fizice de tip cilindru-cap-sector (CHS) la accesarea discului. LILO folosește geometria discului furnizată de BIOS și calculează adresele fizice la rulare, adrese care ar trebui să funcționeze corect.

Periferic SCSI la toți identificatorii SCSI

Dacă un periferic SCSI afișează toți identificatorii SCSI posibili, înseamnă că ați configurat perifericul respectiv cu același identificator SCSI ca și controllerul SCSI (de regulă 7).

Modificați identificatorul perifericului respectiv la o altă valoare. (Multe periferice dispun de un comutator simplu pentru setarea identificatorului SCSI; la multe altele, pentru modificarea identificatorului trebuie schimbat un jumper.)

Periferic SCSI la toate LUN-urile

Dacă un periferic SCSI este afișat la toate numerele de unitate logică (Logical Unit Numbers - LUN), probabil că perifericul are erori de *firmware* - codul încorporat în interfața SCSI a perifericului. Pentru a rezolva aceste erori, mai întâi folosiți următoarea linie de comandă în timpul secvenței de boot:

```
max_scsi_luns=1
```



Secret

Dacă perifericul funcționează cu această opțiune, o puteți adăuga la „lista neagră” a perifericelor SCSI din masivul de structuri denumit `device_list` din fișierul /usr/src/linux/drivers/scsi/scsi.c. Definiția acestei structuri și conținutul curent al masivului sunt prezentate în continuare:

```
struct dev_info{
    const char * vendor;
    const char * model;
    const char * revision; /*Latest version known to be bad. Not used yet
    unsigned flags;
};

/*
 * This is what was previously known as the blacklist.
 * The concept has been expanded so that we can specify other
 * types of things we need to be aware of.
 */
static struct dev_info device_list[] =
{
    {"CHINON", "CD-ROM CDS-431", "H42", BLIST_NOLUN}, /* Locks up if polled for lun
    != 0 */
    {"CHINON", "CD-ROM CDS-535", "Q14", BLIST_NOLUN}, /* Locks up if polled for lun
    != 0 */
    {"DENON", "DRD-25X", "V", BLIST_NOLUN}, /* Locks up if probed for lun
    != 0 */
    {"HITACHI", "DK312C", "CM81", BLIST_NOLUN}, /* Responds to all lun - dtg */
    {"HITACHI", "DK314C", "CR21", BLIST_NOLUN}, /* responds to all lun */
    {"IMS", "CDD521/10", "2.06", BLIST_NOLUN}, /* Locks-up when LUN>0 polled.
    */
    {"MAXTOR", "XT-3280", "PR02", BLIST_NOLUN} /* Locks-up when LUN>0 polled.
    */
    {"MAXTOR", "XT-4380S", "B3C", BLIST_NOLUN} /* Locks-up when LUN>0 polled.
    */
    {"MAXTOR", "XT-1240", "I1.2", BLIST_NOLUN} /* Locks-up when LUN>0 polled.
    */
    {"MAXTOR", "XT-4170S", "B5A", BLIST_NOLUN} /* Locks-up sometimes when
    LUN>0 polled. */
    {"MAXTOR", "XT-8760S", "B7B", BLIST_NOLUN} /* guess what? */
    {"MEDIASIS", "RENO CD-ROMX2A", "2.03", BLIST_NOLUN}, /* Responds to all lun */
    {"MICROP", "4110", "B", BLIST_NOTQ} /* Buggy Tagged Queuing */
    {"NEC", "CD-ROM DRIVE:841", "1.0", BLIST_NOLUN} /* Locks-up when LUN>0 polled.
    */
    {"RODIME", "R03000S", "2.33", BLIST_NOLUN}, /* Locks-up when LUN>0 polled.
    */
    {"SANYO", "CRD-250S", "1.20", "BLIST_NOLUN"}, /* causes failed REQUEST SENSE
    on lun 1 for aha152x controller, which causes SCSI code to reset bus. */
    {"SEAGATE", "ST157N", "004j", BLIST_NOLUN}, /* causes failed REQUEST SENSE
    on lun 1 for aha152x controller, which causes SCSI code to reset bus. */
    {"SEAGATE", "ST296", "921", BLIST_NOLUN}, /* Responds to all lun */
    {"SEAGATE", "ST1581", "6538", BLIST_NOLUN}, /* Responds to all lun */
    {"SONY", "CD-ROM CDU-541", "4.3d", BLIST_NOLUN},
    {"SONY", "CD-ROM CDU-55S", "1.0i", BLIST_NOLUN},
    {"SONY", "CD-ROM CDU-561", "1.7x", BLIST_NOLUN},
    {"TANDBERG", "TDC 3600", "U07", BLIST_NOLUN}, /* Locks up if polled for lun
    != 0 */
    {"TEAC", "CD-ROM", "1.06", BLIST_NOLUN} /* causes failed REQUEST SENSE
    on lun 1 for seagate controller, which causes SCSI code to reset bus. */
    {"TEXEL", "CD-ROM", "1.06", BLIST_NOLUN}, /* causes failed REQUEST SENSE
    on lun 1 for seagate controller, which causes SCSI code to reset bus. */
    {"QUANTUM", "LPSS25S", "3110", BLIST_NOLUN}, /* Locks sometimes if polled
    for lun != 0 */
    {"QUANTUM", "PD1225S", "3110", BLIST_NOLUN}, /* Locks sometimes if polled
    for lun != 0 */
    {"MEDIASIS", "CDR-H93MV", "1.31", BLIST_NOLUN}, /* Locks up if polled for lun
    != 0 */
}
```

```

{"SANKYO", "CP525", "6.64", BLIST_NOLUN},      /* causes failed REQ SENSE,
extra reset */
{"HP", "C1750A", "3226", BLIST_NOLUN},         /* scanjet iic */
{"HP", "C1790A", "", BLIST_NOLUN},             /* scanjet iip */
{"HP", "C2500A", "", BLIST_NOLUN},             /* scanjet iicx */

/*
 * Other types of devices that have special flags.
 */
{"SONY", "CD-ROM CDU-8001", "", BLIST_BORKEN},
{"TEXEL", "CD-ROM", "1.06", BLIST_BORKEN},
{"IOMEGA", "Io20S", "F", "", BLIST_KEY},
{"INSITE", "Floptical F*81", "", BLIST_KEY},
{"INSITE", "I325VM", "", BLIST_KEY},
{"NRC", "MBR-7", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"NRC", "MBR-7.4", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"NAKAMICH", "MJ-4.8S", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"PIONEER", "CD-ROM DRM-602X", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"PIONEER", "CD-ROM DRM-604X", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"EMULEX", "MD21/S2 ESDI", "", BLIST_SINGELUN},
{"CANON", "IPUBJD", "", BLIST_SPARSELUN},
{"MATSUSHITA", "PD", "", BLIST_FORCELUN | BLIST_SINGELUN},
{"YAMAHA", "CDR100", "1.00", BLIST_NOLUN}, /* Locks up if polled for lun != 0 */
{"YAMAHA", "CDR102", "1.00", BLIST_NOLUN}, /* Locks up if polled for lun != 0 */
{"nCIPHER", "fastness Crypto", "", BLIST_FORCELUN},

/*
 * Must be at end of list...
 */
{NULL, NULL, NULL};

```



Din această listă vă puteți face o idee referitoare la tipurile de periferice SCSI care prezintă problema de apariție la toate LUN-urile. Dacă aveți o problemă similară cu un periferic SCSI, puteți adăuga numele dispozitivului respectiv la această listă înainte de ultima linie, acțiune nerecomandată dacă nu cunoașteți limbajul de programare C.

Detecție erori la periferic SCSI fără erori (error-free)

Cauza acestei probleme rezidă într-un cablaj incorect sau o terminație neadecvată. Verificați toate cablurile și verificați dacă există terminatori la ambele capete ale magistralei SCSI.

Probleme cu nucleul la conectare în rețea a perifericului SCSI

Dacă un nucleu Linux cu suport pentru rețea nu funcționează cu periferice SCSI, problema poate fi funcția de autotestare a driverelor de rețea. *Funcția de autotestare* are scopul de a detecta tipul de hardware de rețea în mod automat. Driverelor de rețea citesc din, respectiv scriu la adrese I/O bine determinate în timpul autotestării. Dacă o adresă I/O este aceeași cu aceea folosită de un periferic SCSI, sistemul are o problemă. În acest caz, trebuie să verificați adresa I/O, precum și valorile IRQ și DMA ale plăcilor de rețea și ale controllerului SCSI, pentru a vă asigura că nu există conflicte. Majoritatea controllerelor SCSI (și chiar adaptoarele de rețea) vă permit să configurați acești parametri prin intermediul unui program de configurare livrat o dată cu adaptorul.

Periferic detectat, dar inaccesibil

Dacă nucleul detectează un periferic SCSI (așa cum s-a semnalat în mesajele de boot, care pot fi vizualizate folosind comanda `dmesg | more`) dar nu puteți accesa perifericul, atunci fișierul perifericului lipsește din directorul `/dev`.



Pentru a adăuga fișierul de periferic, intrați în sistem ca root, treceți în directorul `/dev` și apoi folosiți script-ul MAKEDEV pentru a crea fișierul de periferic. Pentru a adăuga un fișier de periferic pentru o unitate de bandă SCSI, de exemplu, veți folosi următoarea comandă:

```

cd /dev
./MAKEDEV st0

```

Când intrați în sistem ca root, directorul curent nu se găsește, în mod obișnuit, în variabila de mediu PATH. De aceea aveți nevoie de prefixul `./` la execuția script-ului MAKEDEV. Puteți afla mai multe informații despre script-ul MAKEDEV folosind următoarea comandă:

```
man MAKEDEV
```

Blocare SCSI

Dacă sistemul SCSI se blochează, verificați placa controller SCSI folosind orice software de diagnostică livrat o dată cu placa (de regulă, acest program rulează sub DOS). Căutați eventuale conflicte de adrese I/O, IRQ sau DMA cu alte plăci. Unele plăci de sunet, de exemplu, folosesc un canal DMA pe 16 biți în afară de un canal DMA pe 8 biți; verificați dacă nu ați folosit din greșeală același canal DMA pe 16 biți pentru placa SCSI.



Secret

Driverul SCSI sub Linux pentru unele plăci SCSI suportă o singură comandă *outstanding* la un moment dat. Cu o asemenea placă SCSI, dacă un periferic precum o unitate de bandă este ocupat cu derularea, sistemul poate fi incapabil de a accesa alte periferice SCSI (cum ar fi un hard-disc sau o unitate CD-ROM) înălțuite cu unitatea de bandă respectivă. O soluție la această problemă poate fi de a adăuga un al doilea controller SCSI pentru unitățile de bandă.

Periferice SCSI care nu au fost găsite

Dacă nucleul Linux nu detectează perifericele SCSI la inițializarea sistemului, veți primi următorul mesaj la încărcarea sistemului de operare Linux:

```
scsi : 0 hosts
```

Dacă ați primit acest mesaj, dar știți că perifericele SCSI există (și funcționează sub DOS), problema poate fi lipsa unui BIOS pe controllerul SCSI; rutinele de autotestare care detectează perifericele SCSI se bazează pe BIOS.

Această problemă apare la următoarele tipuri de plăci SCSI:

- Adaptec 152x, 151x, AIC-6260 și AIC-6360
- Future Domain 1680, TMC-950 și TMC-8xx
- Trantor T128, T128F și T228F
- Seagate ST01 și ST02
- Western Digital 7000



Secret

Chiar dacă un controller SCSI dispune de BIOS, există adesea jumpere pentru dezactivarea BIOS-ului. Chiar dacă ați dezactivat BIOS-ul dintr-un motiv sau altul, poate doriți să îl reactivați (pentru instrucțiuni, citiți documentația plăcii controller SCSI) pentru ca Linux să poată detecta perifericele SCSI în mod automat.

Pentru o placă SCSI, precum Adaptec 151x, care nu dispune de nici un fel de BIOS, folosiți următoarea linie de comandă în timpul execuției secvenței de boot pentru a forța detecția plăcii:

```
Aha152x=0x340,11,7,1
```


Rezumat

Aveți nevoie de un hard-disc pentru a instala Linux pe calculatorul dumneavoastră. În particular, pentru a instala Linux cu succes, controllerul de hard-disc al calculatorului trebuie suportat de Linux. Aceasta nu ar trebui să fie o problemă, deoarece Linux suportă popularele interfețe de tip IDE și Enhanced IDE, precum și interfața SCSI. Acest capitol descrie modul de instalare și utilizare a hard-discurilor în Linux. Din lectura capitolului de față, ați învățat următoarele:

- ▶ Suportul Linux pentru o anumită unitate de hard-disc depinde de controllerul de disc folosit pentru conectarea unității respective la placa de bază a calculatorului. Linux include suport pentru popularele controllere de disc de tip IDE (Integrated Drive Electronics), Enhanced IDE (EIDE) și SCSI pentru magistrale ISA, EISA, VLB și PCI. Aceste controllere de disc acoperă practic toate tipurile de discuri folosite de PC-uri.
- ▶ Trebuie să partiționați un disc pentru a instala Linux și, când instalați Linux, trebuie să creați un spațiu de schimb și să stabiliți un mecanism pentru încărcarea sistemului de operare Linux. Puteți folosi LILO pentru a gestiona procesul de boot, chiar dacă dispuneți de sisteme de operare multiple (cum ar fi DOS și Windows 95) pe disc.
- ▶ Controllerul de disc „vede” discul sub forma unei colecții de sectoare de 512 baiți amplasate pe platouri magnetice montate pe un ax și care se rotesc sub capetele de citire-scriere. Această structură fizică a unui hard-disc permite considerarea unui disc ca o colecție de cilindri, capete și sectoare (CHS). Linux preferă să „vadă” discul sub forma unei succesiuni de sectoare care pot fi adresate secvențial (adresă liniară de bloc sau LBA).
- ▶ Sistemul BIOS al calculatorului folosește adresarea CHS și limitează adresa de cilindru la o valoare pe 10 biți, determinând astfel o limită de 1024 de cilindri pentru geometriile discurilor. Deoarece limitările hardware nu permit existența a mai mult de 16 capete și 63 de sectoare, BIOS-ul poate manipula discuri de maximum 504 MB ($1024 \times 16 \times 63$ sectoare; un sector are 512 baiți). Se folosesc anumite trucuri pentru a manipula discurile mai recente, de mare capacitate, dar aceste trucuri intră uneori în conflict cu modalitatea folosită de Linux pentru adresarea discurilor. În general, se recomandă păstrarea setărilor DOS pentru parametrii discului precum numerele de cilindri, capete și sectoare.
- ▶ SCSI este popular deoarece un singur controller SCSI permite conectarea a maximum șapte periferice. Unicul dezavantaj este costul relativ ridicat al controllerului SCSI în raport cu plăcile de interfață EIDE.
- ▶ Linux include o mare varietate de controllere SCSI, inclusiv popularele Adaptec și BusLogic.
- ▶ Multe calculatoare de producție recentă dispun de magistrala PCI și de cipurile controller EIDE pentru PCI înglobate în placa de bază. S-a observat că două dintre aceste controllere – RZ 1000 și CMD 640 – au erori care survin la conectarea de periferice IDE multiple pe controllerul EIDE. Linux poate rezolva aceste probleme.
- ▶ Cu puțin efort, puteți folosi o unitate Iomega Zip sub Linux.

Capitolul 12

Unități CD-ROM și plăci de sunet

În acest capitol:

- ▶ Veți examina diferitele tipuri de interfețe pentru unități CD-ROM
- ▶ Veți specifica nume de periferice Linux pentru unități CD-ROM și plăci de sunet
- ▶ Veți folosi anumite tipuri de unități CD-ROM în Linux
- ▶ Veți remedia anumite probleme ale unităților CD-ROM în Linux
- ▶ Veți examina tipurile de plăci de sunet folosite în Linux
- ▶ Veți include în Linux suport pentru un anumit tip de placă de sunet
- ▶ Veți testa și depana plăci de sunet sub Linux

Dacă ați instalat sau intenționați să instalați Linux de pe CD-ROM-ul anexat acestei cărți, este foarte posibil ca sistemul dumneavoastră să dispună deja de o unitate CD-ROM. Probabil că citiți acest capitol deoarece aveți întrebări referitoare la utilizarea unității CD-ROM pentru instalarea sistemului Linux; răspunsurile la aceste întrebări se găsesc aici. Ca și în cazul altor echipamente periferice (cum ar fi o unitate de hard-disc), unitatea CD-ROM se conectează la placa de bază a calculatorului dumneavoastră prin intermediul unei plăci controller. Astfel, problema efectivă a utilizării unei unități CD-ROM sub Linux se reduce la includerea sau nu în Linux a unui driver pentru interfața unității CD-ROM respective. Acest capitol va discuta despre diferitele interfețe CD-ROM și le indică pe acelea care sunt suportate în Linux.

Am alăturat unitățile CD-ROM cu plăcile de sunet, deoarece multe unități CD-ROM se conectau la calculator folosind interfețe încorporate în plăcile de sunet. Inițial, distribuitorii livrau unitățile CD-ROM și plăcile de sunet sub forma unui pachet, deoarece aveți nevoie de ambele periferice pentru a vă bucura de programele multimedia care folosesc sunete, imagini și animație. Unitatea CD-ROM pune la dispoziția utilizatorului spațiul necesar pentru stocarea secvențelor video, imaginilor și fișierelor de sunet existente într-o aplicație multimedia obișnuită. Placa de sunet permite calculatorului să genereze sunet de calitate profesională – celălalt ingredient obligatoriu dintr-o aplicație multimedia.

Acest capitol descrie diferite tipuri de unități CD-ROM suportate de Linux, clasificate după tipul de interfață. De asemenea, veți găsi informații privind modul de a include suport pentru diferitele plăci de sunet în Linux.

Unități CD-ROM

Fiecare CD-ROM poate memora maximum 650 MB de date (echivalentul a 450 de dischete de 3,5 inch de înaltă densitate), iar costurile sale de producție sunt reduse. Mediul fizic al unui CD-ROM este același cu cel folosit pentru compact discuri audio (CD-uri), în speță un disc de

policarbonat acoperit cu un strat aluminizat. Un laser citește datele, care sunt codificate în canale microscopice practicate în stratul aluminizat. Mediul CD-ROM este mai robust și mai fiabil decât mediile magnetice, precum dischetele. Toți acești factori transformă CD-ROM-ul într-un mediu atractiv pentru distribuția datelor și a programelor. În realitate, majoritatea cărților de Linux (inclusiv cea de față) conțin un CD-ROM cu o distribuție Linux completă, care include sistemul de operare și o mulțime de programe frecvent folosite.

În lista aceea de calități nemaipomenite ale unui CD-ROM – capacitate ridicată, costuri reduse și fiabilitate – nu este menționată deloc viteza, deoarece ratele de transfer al datelor pentru unitățile CD-ROM nu sunt la fel de ridicate ca în cazul hard-discurilor. La apariția unităților CD-ROM, acestea puteau transfera date la viteze de aproximativ 150 KB pe secundă. Aceste unități erau cunoscute sub numele de unități CD-ROM *single-speed* (cunoscute și sub numele de *1X*). Unitățile CD-ROM dublă viteză (*2X*), care asigurau viteze de transfer de 300 KB pe secundă, au devenit apoi disponibile pe scară largă. În prezent, majoritatea sistemelor includ unități CD-ROM 16X sau 24X, care pot realiza viteze de transfer de maximum 3600 KB pe secundă și timpi medii de acces de 95 milisecunde (pentru comparație, timpul de acces al unui hard-disc este de aproximativ 10 milisecunde).



Observație

Majoritatea unităților CD-ROM conțin informații într-un sistem de fișiere ISO-9660 (cunoscut anterior sub numele de High Sierra). Acest sistem de fișiere conține numai nume de fișiere de tip 8.3 (în stil MS-DOS), precum README.TXT, cu un nume compus din maximum opt caractere și o extensie opțională din trei caractere. O extensie a sistemului de fișiere ISO-9660, cunoscută sub numele de Rock Ridge Extensions, folosește câmpuri nefolosite pentru suportul de nume de fișiere mai lungi și atribute de fișiere stil UNIX, precum proprietarul și legăturile simbolice.

Majoritatea unităților CD-ROM mai permit derularea de CD-uri audio prin intermediul unei mufe de cască externă. Există și o linie de ieșire prin care vă puteți conecta la placa de sunet, deci puteți derula un CD audio folosind difuzoarele atașate plăcii de sunet.

Unități CD-ROM suportate

Ca și în cazul hard-discurilor, suportul Linux pentru o unitate CD-ROM depinde de interfața prin care unitatea CD-ROM se conectează la placa de bază a calculatorului. Unitățile CD-ROM dispun de trei categorii de interfețe:

- **AT Attachment Packet Interface (ATAPI).** ATAPI este o specificație recentă pentru accesarea și controlul unei unități CD-ROM conectate la calculator prin intermediul AT Attachment (ATA). ATAPI câștigă tot mai multă popularitate, deoarece este construită pe interfața IDE, mai puțin costisitoare. (ATA este noul nume al IDE.)
- **Small Computer System Interface (SCSI).** SCSI este popular datorită vitezelor de transfer relativ mari și posibilității de a suporta un număr mare de periferice. Unicul dezavantaj este că aveți nevoie de o placă controller SCSI relativ costisitoare pentru calculator.
- **Interfețe brevetate.** Multe firme producătoare de unități CD-ROM și-au realizat propria interfață patentată între unitatea CD-ROM și placa de bază a calculatorului. Multe plăci de sunet includ o unitate CD-ROM încorporată, care de regulă este o interfață brevetată. Problema interfețelor patentate este necesitatea unui driver Linux pentru fiecare asemenea interfață în parte (spre deosebire de un driver SCSI, de exemplu, pentru accesarea oricărui periferic SCSI).

Unități CD-ROM ATAPI

ATA (AT Attachment) este numele standard ANSI (Institutul Național pentru Standarde al SUA) pentru interfața IDE folosită la conectarea unităților de hard-disc la calculatoare. ATAPI (ATA Packet Interface) este un protocol (similar cu SCSI) pentru controlul mediilor de stocare cum sunt unitățile CD-ROM și unitățile de bandă. Deși ATAPI este relativ nou, devine cu rapiditate cel mai popular tip de interfață pentru unitățile CD-ROM. Motivul este că ATAPI este bazat pe ATA (sau IDE) și nu necesită un controller sau un cablu costisitor. De asemenea, un CD-ROM ATAPI se poate conecta pur și simplu ca a doua unitate pe aceeași interfață unde este conectată unitatea de disc a calculatorului. Aceasta înseamnă că unitatea CD-ROM ATAPI nu necesită o placă de interfață separată.

Nucleul Linux include un driver ATAPI care este compatibil cu orice unitate CD-ROM ATAPI. Unitățile CD-ROM ATAPI sunt produse de mai multe firme, cum sunt Aztech, Mitsumi, NEC, Philips, Sony sau Toshiba. În prezent, majoritatea calculatoarelor (cum sunt cele produse de Gateway sau Dell) sunt configurate pentru unități CD-ROM ATAPI.

Unități CD-ROM SCSI

Linux include suport pentru o unitate CD-ROM SCSI conectată la una dintre plăcile controller SCSI suportate (citiți capitolul 11 pentru mai multe informații). Unica restricție este dimensiunea blocului (pentru transferul de date) a unității CD-ROM SCSI, care trebuie să fie de 512 sau 2048 de baiți, care caracterizează marea majoritate a unităților CD-ROM existente pe piață.

Unele unități CD-ROM includ un controller cu o interfață modificată, care nu este complet compatibilă SCSI. Această interfață este, în esență, patentată; unitățile CD-ROM cu asemenea interfețe nu pot fi folosite cu un driver SCSI.

Unitățile CD-ROM SCSI sunt produse de firme precum Plextor, Sanyo și Toshiba.

Unități CD-ROM brevetate

Deși unitățile CD-ROM ATAPI și SCSI se încadrează în categorii bine definite și sunt agreate de Linux, situația se complică în momentul utilizării unităților CD-ROM cu interfață patentată. Iată care sunt motivele de confuzie:

- Unii producători, cum este Creative Labs (creatorii faimosului Sound Blaster) au comercializat unități CD-ROM cu toate tipurile de interfețe (ATAPI, SCSI și patentată) pe placa de sunet. Astfel, numele producătorului în sine nu are nici o semnificație; trebuie să cunoașteți care este tipul de interfață folosit de unitatea CD-ROM.
- Uneori, distribuitorii de calculatoare încadrează interfața cu unitatea CD-ROM în categoria IDE, deși în realitate interfața este patentată. Ca și interfețele IDE (sau ATAPI), interfața patentată cu unitatea CD-ROM este necostisitoare și populară. Deseori, o interfață CD-ROM patentată este incorect etichetată ca interfață IDE.

După cum probabil v-ați dat seama, interfețele patentate cu unitățile CD-ROM sunt populare datorită tendinței de a fi mai simple în comparație cu SCSI, care a reprezentat prima alternativă la interfețele patentate înainte de apariția ATAPI. Datorită popularității interfeței ATAPI relativ ieftine, majoritatea calculatoarelor noi nu folosesc interfețe patentate pentru unitățile CD-ROM. Deoarece o interfață CD-ROM patentată poate fi totuși încorporată într-o placă de sunet la costuri reduse, unii utilizatori Linux vor dispune de calculatoare cu unități CD-ROM cu interfață brevetată.

Tabelul 12.1 prezintă unități CD-ROM cu interfețe patentate, precum și numele driverelor necesare pentru suportul acestor unități.

Tabelul 12.1 Unități CD-ROM cu interfețe patentate

Driver	Unitate CD-ROM
aztcd	Aztech CDA268, Orchid CDS-3110, Okano/Wearnes CDD-110, Conrad TXC, CyCDROM CR520ie/CR940ie
cdu31a	Sony CDU31A/CDU33A
cm206	Philips/LMS CM 206
gscd	GoldStar R420
isp16	Unități CD-ROM atașate la interfață pe o placă de sunet ISP16, MAD16 sau Mozart
mcd	Mitsumi CRMCM LU005S, FX001
optcd	Optics Storage Dolphin 8000AT, Lasermate CR328A
sbpcd	Matsushita/Panasonic (Panasonic CR-521, CR-522, CR-523, CR-562 și CR-563), Kotobuki, Creative Labs (CD-200), Longshine LCS-7260, Teac CD-55A
sjcd	Sanyo H94A
sonycd535	Sony CDU-535/CDU-531

Depanarea unităților CD-ROM

Pentru a instala Linux de pe CD-ROM-ul anexat acestei cărți trebuie să dispuneți de o unitate CD-ROM funcțională sub Linux. Nucleul Linux inițial conține modulele driver CD-ROM pentru toate unitățile CD-ROM suportate, deci dacă unitatea dumneavoastră CD-ROM este suportată de Linux nu ar trebui să aveți nici un fel de probleme. Nu uitați că interfața cu unitatea CD-ROM este ceea ce contează, nu numele mărcii. De exemplu, toate mărcile de unități CD-ROM cu interfața IDE vor funcționa sub Linux, deoarece Linux include suport pentru interfața IDE.

Dacă Linux nu pare să recunoască unitatea CD-ROM după regenerarea nucleului, încercați următoarea succesiune de etape pentru a remedia problema:

1. Dacă ați regenerat nucleul cu suport pentru unitatea dumneavoastră CD-ROM, verificați dacă într-adevăr rulați noua variantă de nucleu. Pentru a vedea numărul de versiune, folosiți comanda `uname -a`. Iată o ieșire specifică comenzii specificate anterior:

```
Linux localhost 2.0.32 #1 Fri Jan 30 07:08:01 EST 1998 i586 unknown
```

Acest rezultat al comenzii `uname` prezintă numărul de versiune al nucleului, ca și data la care a fost generat nucleul. Dacă această dată nu corespunde datei la care ați regenerat nucleul, înseamnă că nu rulați noul nucleu. Parcurgeți etapele descrise în capitolul 2 și verificați dacă într-adevăr ați instalat noul nucleu. O altă problemă comună este când uităm să reîncărcăm sistemul de operare, deci încercați și aceasta.

2. Examinați conținutul fișierului `/proc/devices` pentru a confirma prezența perifericului CD-ROM. Folosiți următoarea procedură pentru a vizualiza conținutul unui fișier `/proc/devices`:

```
cat /proc/devices
Character devices:
```

```
1 mem
2 tty
3 tty
4 tty
5 cua
7 vcs
10 misc
```

Block devices:

```
1 ramdisk
2 fd
3 ide0
7 loop
8 sd
9 md
22 ide1
```

Acest listing corespunde perifericelor instalate în sistemul meu. Trebuie să căutați perifericul CD-ROM în lista de dispozitive bloc. Eu știu că unitatea mea CD-ROM este conectată la interfața IDE secundară. Cele două interfețe IDE corespund perifericelor `ide0` și `ide1` din `/proc/devices`. Deoarece `ide1` apare în listing, știu că driverul CD-ROM se află în nucleu.

Dacă aveți o unitate CD-ROM conectată la o interfață Sound Blaster Pro sau compatibilă, căutați un număr de periferic 25 și un nume de periferic `sbpcd`.

Dacă nu aveți un periferic care corespunde unității dumneavoastră CD-ROM, înseamnă că nu ați configurat nucleul în mod adecvat pentru a include driverul CD-ROM. Reconfigurați și regenerați nucleul, asigurându-vă că ați inclus suport pentru unitatea CD-ROM.

3. Verificați dacă driverul CD-ROM a detectat unitatea CD-ROM la lansarea sistemului. Folosiți comanda `dmesg | more` pentru a examina mesajele de boot și a vedea dacă există o linie care să arate că unitatea CD-ROM a fost găsită. Pe sistemul meu, care dispune de o unitate ATAPI CD-ROM, mesajul arată astfel:

```
hdc: FX400_02, ATAPI, CDROM drive
```

Dacă nu găsiți nici un mesaj de boot despre unitatea CD-ROM, verificați dacă unitatea este fizic instalată. Pentru o unitate CD-ROM externă, verificați dacă unitatea este alimentată și dacă cablurile sunt conectate. Verificați identificatorul și jumperele fiecărei unități și verificați dacă sunt corect setate. Mai întâi, trebuie să verificați dacă unitatea CD-ROM funcționează sub DOS. Dacă ați confirmat acest lucru, atunci unitatea este funcțională sub aspect fizic. În continuare, verificați dacă ați regenerat nucleul cu suport pentru interfața corectă cu unitatea CD-ROM.

4. Verificați dacă puteți citi din unitatea CD-ROM. Încercați comanda prezentată în continuare și observați dacă indicatorul luminos de activitate a unității se aprinde (așa ar trebui) și dacă apar mesaje de eroare (nu ar trebui):

```
dd bs=1024 count=500 < /dev/cdrom > /dev/null
5000+0 records in
5000+0 records in
```

Perifericul `dev/null` este ceea ce se poate numi *bit bucket* (găleată cu biți). Ieșirea direcționată spre `/dev/null` pur și simplu dispare.

Dacă comanda `dd` nu a avut efect, fișierul de periferic pentru unitatea CD-ROM poate fi incorect configurat. Folosiți comanda `ls -l` pentru a vizualiza informații detaliate despre perifericul dumneavoastră CD-ROM. Pentru o unitate CD-ROM ATAPI pe interfața IDE secundară, de exemplu, vom examina `/dev/hdc` după cum urmează:

```
ls -l /dev/hdc
brw-rw--- 1 root disk 22, 0 Feb 4 1995 /dev/hdc
```

Observați numerele 22 și 0, care reprezintă numerele perifericelor principal și secundar. Aceste numere trebuie corectate. (În secțiunile viitoare ale acestui capitol vom prezenta numerele de dispozitiv pentru anumite unități CD-ROM.)

5. Verificați dacă puteți monta unitatea CD-ROM. Inserați un CD-ROM de calitate (cum este cel din această carte) în unitatea CD-ROM și încercați să montați folosind următoarea comandă:

```
mount -t iso9660 -r /dev/cdrom /mnt/cdrom
```

Dacă puteți citi din unitatea CD-ROM cu comanda dd, dar nu puteți monta unitatea, înseamnă că ați configurat nucleul fără suport pentru sistemul de fișiere ISO-9660. Pentru a verifica sistemele de fișiere suportate la un moment dat, folosiți următoarea comandă:

```
cat /proc/filesystems
ext2
msdos
nodev proc
nodev smbfs
iso9660
```

Dacă nu vedeți în listă iso9660, va trebui să regenerați nucleul și să adăugați suport pentru sistemul de fișiere ISO-9660.

6. Dacă nimic nu pare să dea rezultate, puteți citi cel mai recent document CDRM-HOWTO. Pentru a citi documentele HOWTO, tastați `cd /usr/doc/HOWTO` pentru a comuta în directorul unde se află localizate fișierele HOWTO. Apoi tastați `zcat CDRM-HOWTO.gz | more` pentru a vizualiza fișierul CDRM-HOWTO.

Dacă nu reușiți nici acum să determinați unitatea CD-ROM să funcționeze sub Linux, încercați să inserați un anunț într-unul din grupurile de discuții comp.os.linux.

În secțiunile următoare vom prezenta sugestii pentru alte câteva probleme frecvent întâlnite.

Configurarea nucleului pentru anumite unități CD-ROM

Dacă regenerați nucleul și dispuneți de o unitate CD-ROM ATAPI, trebuie să răspundeți afirmativ la următoarele întrebări în timp ce configurați nucleul (în capitolul 2 se află mai multe informații privind configurarea nucleului):

```
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
(CONDIF_BLK_DEV_IDE) [Y/n/?] y
Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/n/?] z
```

Răspunsul prestabilit este dat între paranteze; y înseamnă Yes (Da), iar n înseamnă Nu. Puteți accepta valoarea prestabilită prin apăsarea tastei Enter.



Pentru unități CD-ROM SCSI, răspundeți afirmativ la următoarele întrebări:

Sugestie

```
*
* SCSI support
*
SCSI support (CONFIG SCSI) [N/y/m/?] y
*
* SCSI support type (disk, tape, CD-ROM)
*
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [N/y/m/?] (NEW) n
```

Desigur, mai trebuie să specificați tipul controllerului SCSI; în caz contrar, unitatea CD-ROM SCSI nu va funcționa. Dacă dispuneți de un controller SCSI Adaptec AHA1542, de exemplu, răspundeți afirmativ la următoarea întrebare:

```
Adaptec AHA1542 support (CONFIG SCSI_AHA1542) [N/y/m/?] (NEW) y
```



Sugestie

Dacă unitatea dumneavoastră CD-ROM dispune de o interfață patentată, trebuie să activați suportul pentru interfața respectivă cu unitatea CD-ROM. În continuare sunt prezentate câteva întrebări din versiunea 2.0.31 a nucleului:

```
*
* CD-ROM drivers (not for SCSI or IDE/ATAPI drives)
*
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [Y/n/?]
Aztech/Orchid/Okano/Wearnes/TXC/CyCDROM support (CONFIG_AZTCD) [N/y/m/?]
Goldstar R420 CDROM support (CONFIG_GSCD) [N/y/m/?]
Matsushita/Panasonic/Creative, Longshine/ TEAC CDROM support (CONFIG_SBPCD)
[N/y/m/?]
Mitsumi (standard) [no XA multisession] CDROM support (CONFIG_MCD) [N/y/m/?]
Mitsumi [XA/MultiSession] CDROM support (CONFIG_MCDX) [N/y/m/?]
Optics Storage DOLPHIN 8000AT CDROM support (CONFIG_OPTCD) [N/y/m/?]
Philips/LMS CM206 CDROM support (CONFIG_CM206) [N/y/m/?]
Sanyo CDR-H94A CDROM support (CONFIG_SJCD) [CONFIG_SJCD] [N/y/m/?]
Soft configurable cdrom interface card support (CONFIG_CDI_INIT) [N/y/?]
Sony CDU31A/CDU33A CDROM support (CONFIG_CDU31A) [N/y/m/?]
Sony CDU535 CDROM support (CONFIG_CDU535) [N/y/m/?]
```

Răspundeți afirmativ la întrebarea despre unitatea CD-ROM existentă în calculatorul dumneavoastră. Dacă dispuneți de o unitate CD-ROM Mitsumi (cu o interfață patentată, nu ATAPI), de exemplu, tastați y ca răspuns la linia care începe cu Mitsumi (standard).

Versiunile mai recente ale nucleului Linux pot suporta și alte tipuri de interfețe patentate cu unități CD-ROM. Verificați cu atenție prompturile înainte de a răspunde la întrebările puse de programul de configurare a nucleului.



Sugestie

```
ISO9660 cdrom filesystem support (CONFIG_ISO9660_FS) [M/n/y/?] y
```

Probleme cu unitățile CD-ROM IDE (ATAPI)

Când un calculator are o interfață CD-ROM ATAPI, adaptorul său ATA (IDE) are două interfețe: primară și secundară, fiecare interfață fiind capabilă de a suporta două unități. În Linux, cele două periferice primare sunt /dev/hda și /dev/hdb, care sunt folosite de obicei pentru unitățile de hard-disc. Perifericele IDE secundare sunt /dev/hdc și /dev/hdd. Astfel, dacă aveți o unitate de disc IDE conectată la interfața primară și o unitate CD-ROM IDE conectată la interfața secundară, unitatea de disc va fi /dev/hda, iar unitatea CD-ROM va fi /dev/hdc.



Din cele două periferice ale unei interfețe, una este concepută ca master și a doua ca slave. Când un singur periferic IDE este atașat la o interfață, acesta trebuie desemnat ca master (sau single). De regulă, perifericul IDE are un jumper (conector care leagă o pereche de pini) pentru a specifica dacă perifericul este master sau slave.

Când o singură unitate CD-ROM IDE este conectată la interfața secundară, trebuie să setați jumperul de pe unitatea CD-ROM pentru a o transforma în master. Dacă ați achiziționat un

calculator cu o unitate CD-ROM IDE preinstalată, acest jumper este probabil deja setat. Dacă Linux refuză să recunoască unitatea CD-ROM IDE, trebuie să examinați parametrii unității CD-ROM, verificând dacă aceasta folosește interfața IDE secundară (IRQ 15 și adresa I/O 170H) și dacă este stabilită ca master.



Secret

Unele plăci suportă mai mult de două interfețe IDE. În acest caz, trebuie să știți că unitatea CD-ROM IDE Linux poate să nu recunoască decât interfețele primară și secundară. Creative Labs Sound Blaster 16 este în mod prestabilit a patra interfață. Dacă conectați o unitate CD-ROM la Sound Blaster 16 în vederea utilizării sub Linux, trebuie să setați jumperii de la placa de sunet, deci interfața IDE este interfața secundară, și nu a patra.

Parametri de boot pentru unități CD-ROM

În mod normal, driverele Linux găsesc unitățile CD-ROM (și alte periferice) prin *probare*, adică prin citire din și scriere la diferite adrese I/O. De asemenea, driverele Linux folosesc toate informațiile existente în memoria CMOS a calculatorului. CMOS este un semiconductor metal-oxid complementar, adică un tip de semiconductor. Fiecare calculator dispune de o mică cantitate de spațiu de memorie CMOS non-volatilă alimentată cu baterii, unde se află stocate informații cruciale despre calculator, cum ar fi numărul tipurilor de unități de disc. Ceasul de timp real este de asemenea stocat în CMOS.



O problemă a probării este aceea că necesită citirea din sau scrierea la anumite adrese I/O. În funcție de perifericul care folosește adresa I/O, probarea poate determina lipsa de responsivitate a sistemului. Când driverul Sound Blaster Pro CD, sbpcd, probează o unitate CD-ROM, acesta poate accesa o adresă I/O folosită într-o placă Ethernet NE2000. Dacă se întâmplă acest lucru, sistemul devine neresponsiv.

Pentru a evita probarea, puteți transmite parametrii de periferic specifici pentru dispozitivul corespunzător unității dumneavoastră CD-ROM la încărcarea sistemului de operare.

Dacă dispuneți de o unitate CD-ROM cu interfața Sound Blaster Pro CD, numele perifericului este sbpcd și puteți introduce valorile parametrilor la prompt-ul de boot LILO după cum urmează: sbpcd=0x230, SoundBlaster

Această comandă indică driverului sbpcd că adresa I/O a unității CD-ROM este 230H. Parametrii exacti de boot ai unui periferic depind de driverul de periferic respectiv. În secțiunile viitoare vom descrie parametrii de boot specifici pentru driverele de periferic CD-ROM.



Dacă folosiți LILO, puteți insera parametrii de boot în fișierul /etc/lilo.conf, după cum urmează:

```
= "sbpcd=0x230, SoundBlaster"
```

Numele perifericelor CD-ROM

Linux folosește un nume de periferic unic pentru fiecare tip de interfață CD-ROM. Perifericele CD-ROM sunt dispozitive bloc similare perifericului disc, precum /dev/hda și /dev/sda. În tabelul 12.2 sunt prezentate numele perifericelor CD-ROM.

Tabelul 12.2 Numele de dispozitiv ale unităților CD-ROM în Linux

Nume de dispozitiv	Tip unitate CD-ROM
/dev/aztcd	Interfață unitate CD-ROM Aztech

Nume de dispozitiv	Tip unitate CD-ROM
/dev/cdu31a	Interfață unitate CD-ROM Sony CDU31A/CDU33A
/dev/cm206cd	Interfață unitate CD-ROM Philips/LMS
/dev/gscd	Interfață unitate CD-ROM GoldStar
/dev/hdc	Unitate CD-ROM ATAPI pe interfața IDE secundară pe un controller EIDE
/dev/mcd	Interfață unitate CD-ROM Mitsumi
/dev/optcd	Interfață unitate CD-ROM Optics Storage
/dev/sbpcd	Interfață unitate CD-ROM Sound Blaster Pro
/dev/sjcd	Interfață unitate CD-ROM Sanyo
/dev/sonycd35	Interfață unitate CD-ROM Sony CDU535/CDU-531

Prin convenție, perifericul CD-ROM generic /dev/cdrom este setat ca o legătură la perifericul CD-ROM efectiv existent în sistemul dumneavoastră. Pe sistemul meu, care dispune de o unitate CD-ROM ATAPI, un listing detaliat al /dev/cdrom se prezintă astfel:

```
ls -l /dev/cdrom
lrwxrwxrwx 1 root root 3 Mar 11 15:15 /dev/cdrom -> hdc
```

Ieșirea comenzii ls arată că /dev/cdrom este o legătură simbolică spre /dev/hdc, care este dispozitivul IDE de tip bloc. (Același nume de periferic este folosit pentru un disc sau o unitate CD-ROM conectată la interfață.)

Pe de altă parte, pe vechiul meu PC 386, unitatea CD-ROM se găsește pe o placă Sound Blaster Pro. În acest caz, listingul detaliat al /dev/cdrom prezintă următoarele:

```
ls -l /dev/cdrom
lrwxrwxrwx 1 root root 10 Sep 29 20:19 /dev/cdrom -> sbpcd
```



Convenția de definire a unei legături simbolice între perifericul generic /dev/cdrom și unitatea CD-ROM efectiv existentă în sistem implică faptul că programele care folosesc unitatea CD-ROM să poată face referire la aceasta sub numele /dev/cdrom, fără a ține cont de interfața reală cu unitatea. Datorită legăturii simbolice între /dev/cdrom și perifericul CD-ROM real, toate cererile de intrare sau ieșire ajung direct la driverul CD-ROM, care știe cum să manipuleze respectivele cereri.

Utilizarea unității CD-ROM sub Linux

Dacă introduceți un CD-ROM nou în unitate, trebuie să montați discul înainte de a-l putea folosi. Pentru a monta discul, folosiți următoarea comandă:

```
mount -t iso9660 -r /dev/cdrom /mnt/cdrom
```

Opțiunea -t arată că sistemul de fișiere al unității CD-ROM este ISO-9660 (fapt adevărat pentru o unitate CD-ROM obișnuită), iar opțiunea -r arată că unitatea CD-ROM este montată ca read-only.



Sugestie

Dacă doriți să extrageți CD-ROM-ul din unitate și să încărcați unul nou, mai întâi trebuie să demontați sistemul de fișiere al discului prin folosirea comenzii umount.

Tastați umount /mnt/cdrom.

Apoi puteți extrage discul din unitate prin apăsarea butonului Eject al unității.



Atenție!

Când folosiți medii amovibile de stocare a datelor, cum sunt discurile CD-ROM sau unitățile amovibile (chiar și dischetele) folosiți întotdeauna comanda `umount` înainte de a extrage discheta sau CD-ROM-ul amovibil. Pentru dischete, trebuie să demontați discheta numai dacă ați montat-o anterior.

Derularea de CD-uri audio în unitatea CD-ROM

Pentru a derula CD-uri audio în unitatea CD-ROM aveți nevoie de o aplicație specială. CD-ROM-ul anexat acestei cărți are inclusă aplicația `xplaycd`, o aplicație X care furnizează un panou de control grafic pentru derularea CD-urilor audio.



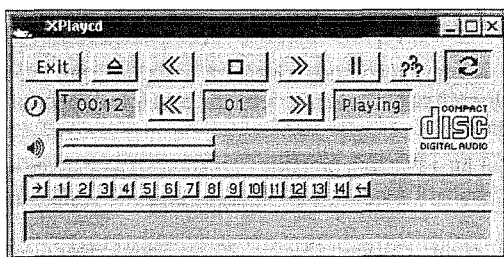
Sugestie

Înainte de a folosi programul `xplaycd`, verificați dacă ați demontat orice CD-ROM aflat în unitate (folosiți comanda `umount /dev/cdrom`) și introduceți în unitate un CD audio.

Programul `xplaycd` necesită sistemul X Window System pentru a putea rula, deci mai întâi trebuie să lansați X și apoi să rulați `xplaycd` la promptul interpretorului dintr-o fereastră xterm. Figura 12.1 prezintă interfața grafică utilizator a aplicației `xplaycd`.

Figura 12-1

Panoul grafic de control al programului `xplaycd` pentru derularea CD-urilor audio.



Întăitor

Dacă doriți să intrați în sistem ca utilizator normal și să derulați CD-uri audio în unitatea CD-ROM, mai întâi trebuie să intrați ca `root` și să setați permisiunile pentru unitatea CD-ROM, pentru ca orice utilizator să poată citi din unitatea respectivă. Trebuie să setați permisiunea pentru unitatea CD-ROM reală, nu pentru perifericul generic `/dev/cdrom`. Pentru a seta parametrii de permisiune, parcurgeți următoarele etape:

1. Intrați în sistem ca `root`. Dacă vă găsiți deja în sistem, folosiți comanda `su` pentru a vă asuma identitatea utilizatorului `root`.

2. Determinați numele real al perifericului CD-ROM. Tastați următoarea comandă:

```
ls -l /dev/cdrom
lrwxrwxrwx 1 root root 3 Mar 11 15:15 /dev/cdrom -> hdc
```

În listingul anterior, veți descoperi numele real al perifericului imediat după simbolul `->` din listing. În cazul de mai sus, numele real al unității CD-ROM este `/dev/hdc`.

3. Permiteți citirea perifericului real de către toți utilizatorii folosind comanda `chmod`, după cum urmează:

```
chmod o+r /dev/hdc
```

4. Verificați setările de permisiune ale unității CD-ROM efective. Permisiunile pentru unitatea `/dev/hdc` se verifică după cum urmează:

```
ls -l /dev/hdc
```

```
brw-rw-r-- 1 root disk 22, 0 Feb 4 1995 /dev/hdc
```

Observați ultimele trei caractere (`-r-`) din prima coloană, prin care se arată că toți utilizatorii au acces de citire la periferic. Mai observați litera `b` de la începutul liniei, care arată că perifericul este un periferic de tip bloc (într-un listing lung, perifericele de tip caracter au prima literă `c`).

Informații caracteristice unității CD-ROM

În secțiunile următoare vom rezuma informațiile specifice anumitor drivere de periferic. Aceste informații includ numele fișierului de dispozitiv (cum ar fi `/dev/sbpcd` pentru o unitate CD-ROM cu interfață Sound Blaster Pro), toți parametrii de boot acceptați de driver și toate posibilitățile caracteristice ale driverului.

Trebuie să citiți secțiunea care se referă la interfața CD-ROM a unității existente în sistemul dumneavoastră.

Driverul `scd`

Driverul `scd` include suport pentru unitățile CD-ROM care se conectează prin intermediul plăcilor controller SCSI. Driverul `scd` are următoarele caracteristici:

```
Allows multiple CD-ROM drives: yes
Supports loadable module: yes
Can read audio frames: no
Performs auto-probing: yes
Name of device file: /dev/scd0, major device number = 11
Device configuration file: /usr/include/linux/cdrom.h
Kernel option during make config: SCSI CDROM support?
Name of README file: none
```



Secret

Driverul `scd` include suport pentru unități CD-ROM multiple pe magistrala SCSI. Numele fișierelor de dispozitiv sunt `/dev/scd0`, `/dev/scd1`, `/dev/scd2` etc., cu numerele minore corespunzătoare 0, 1, 2 etc. Toate perifericele au același număr major, în speță 11.

Driverul CD-ROM IDE

Driverul de periferic IDE suportă toate tipurile de periferice IDE, care variază de la unități de hard-disc la unități CD-ROM. Driverul găsește unitățile prin autoprobare și atribuie nume de dispozitiv succesive: `/dev/hda` și `/dev/hdb` pentru unitățile de pe interfața IDE primară, respectiv `/dev/hdc` și `/dev/hdd` pentru unitățile de pe interfața IDE secundară. O unitate CD-ROM situată pe interfața IDE secundară are numele de dispozitiv `/dev/hdc` (deoarece unitatea CD-ROM este mai lentă decât hard-discul, unitățile CD-ROM sunt conectate la o interfață IDE separate, nefiind setate ca unități slave pe interfața IDE primară).

Driverul CD-ROM IDE are următoarele caracteristici:

```
Allows multiple CD-ROM drives: yes
Supports loadable module: no
Can read audio frames: yes (on some drives)
Performs auto-probing: yes
Name of device file: /dev/hdc or /dev/hdd, major device number = 22
Device configuration file: /usr/include/linux/cdrom.h
Kernel option during make config: Include IDE/ATAPI CDROM support?
Name of README file: /usr/src/linux/Documentation/cdrom/ide-cd
```

Driverul CD-ROM IDE acceptă următorul parametru de boot:

```
hdc=cdrom
```


Acest parametru indică driverului IDE că primul periferic conectat la interfața IDE secundară este o unitate CD-ROM. Dacă mai dispuneți de o altă unitate CD-ROM pe interfața respectivă, puteți include următorul parametru de boot:

```
hdd=cdrom
```

Același driver IDE este folosit atât pentru unități de hard-disc IDE, cât și pentru unități CD-ROM IDE (ATAPI). Parametrii de boot sunt diferiți pentru o unitate de hard-disc.

Driverul sbpcd

Driverul sbpcd include suport pentru o varietate de unități CD-ROM care se conectează la PC prin intermediul unei interfețe care se conformează interfeței patentate folosite la plăcile de sunet Creative Labs Sound Blaster Pro.

Driverul sbpcd are următoarele caracteristici:

```
Allows multiple CD-ROM drives: yes
Supports loadable module: yes
Can read audio frames: yes (CR-562, CR-563, CR-200 only)
Performs auto-probing: yes
Name of device file: /dev/sbpcd, major device number = 25
Device configuration file: /usr/include/linux/sbpcd.h
Kernel option during make config: Matsushita/Panasonic/ Creative, Longshine, TEAC
CDROM support?
Name of README file: /usr/src/linux/Documentation/cdrom/sbpcd
```

Driverul sbpcd acceptă un parametru de boot în formatul următor:

```
sbpcd=IOPORT,interface-type
```

IOPORT este adresa de port a perifericului în format hexazecimal (cum ar fi 0x230).

interface-type este un cuvânt unic, care identifică tipul unității CD-ROM și care trebuie să fie SoundBlaster, SoundMate sau SPEA.

Dacă fișierul de periferic */dev/sbpcd* nu există, creați-l folosind următoarea comandă *mknod*:

```
mknod /dev/sbpcd b 25 0
```

Ultimii trei parametri specifică tipul de periferic (b de la dispozitiv de tip bloc), numărul major de periferic (25 este numărul atribuit pentru */dev/sbpcd*) și numărul minor de periferic (0 pentru prima unitate CD-ROM de pe controller).



Secret

Puteți *înlănțui* (conectați primul la al doilea, al doilea la al treilea etc.) maximum patru unități pe fiecare controller. Folosiți numerele de periferic de la 1 la 3 pentru următoarele trei unități pe care le înlanțuiți la prima unitate.

Driverul sonycd535

Driverul sonycd535 include suport pentru unitățile CD-ROM Sony CDU-535 și CDU-531, care se conectează la calculator prin intermediul plăcii de interfețe brevetate Sony.

Driverul sonycd535 are următoarele caracteristici:

```
Allows multiple CD-ROM drives: no
Supports loadable module: yes
Can read audio frames: no
Performs auto-probing: no
Name of device file: /dev/sonycd535, major device number = 24
Device configuration file: /usr/include/linux/sonycd535.h
Kernel option during make config: Sony CDU535 CDROM support?
Name of README file: /usr/src/linux/Documentation/cdrom/sonycd535
```

Acest driver acceptă următorul parametru de boot:

```
sonycd535=IOPORT
```

IOPORT este adresa portului I/O, scrisă în format hexazecimal (precum 0x320).

Driverul aztcd

Driverul aztcd include suport pentru unitățile CD-ROM Aztech CDA268, Orchid CDS-3110, Okano/Wearnes CDD-110, care folosesc o interfață patentată. Alte unități CD-ROM produse de aceste firme, cum este Aztech CDA269, folosesc interfața IDE. Pentru aceste unități trebuie să folosiți interfața CD-ROM IDE. Driverul aztcd este destinat numai interfețelor patentate.

Driverul aztcd are următoarele caracteristici:

```
Allows multiple CD-ROM drives: no
Supports loadable module: yes
Can read audio frames: no
Performs auto-probing: no
Name of device file: /dev/aztcd, major device number = 29
Device configuration file: /usr/include/linux/aztcd.h
Kernel option during make config: Aztech/Orchid/Okano/Wearnes/TXC CDROM support?
Name of README file: /usr/src/linux/Documentation/cdrom/aztcd
```

Acest driver acceptă următorul parametru de boot:

```
Aztcd=IOPORT
```

IOPORT este adresa portului I/O, scrisă în format hexazecimal (precum 0x340).

Driverul mcd

Driverul mcd include suport pentru unitățile CD-ROM Mitsumi CRMC-LU005S și CRMC-FX001 care dispun de interfață patentată Mitsumi. Multe calculatoare de dată recentă folosesc unitățile CD-ROM Mitsumi, dar nu și interfața patentată. În schimb, tendința actuală este de a folosi driverul Mitsumi cu interfața IDE (ATAPI), deci trebuie să fiți atent când selectați driverul mcd ca driver CD-ROM. Dacă sistemul dumneavoastră dispune de un driver CD-ROM Mitsumi, mai degrabă este o unitate ATAPI decât una cu interfață patentată Mitsumi.

Driverul mcd are următoarele caracteristici:

```
Allows multiple CD-ROM drives: no
Supports loadable module: yes
Can read audio frames: no
Performs auto-probing: no
Name of device file: /dev/mcd, major device number = 23
Device configuration file: /usr/include/linux/mcd.h
Kernel option during make config: Mitsumi (standard) [no XA Multisession] CDROM support?
Name of README file: /usr/src/linux/Documentation/cdrom/mcd
```

Acest driver acceptă următorul parametru de boot:

```
mcd=IOPORT, IRQ
```

IOPORT este adresa portului I/O al plăcii de interfață Mitsumi, scrisă în format hexazecimal (precum 0x340), iar *IRQ* este numărul cererii de întrerupere folosită de placa respectivă.



Sugestie

Mitsumi [XA/MultiSession] CDROM support?

Linux mai include un driver Mitsumi, *mcxd*, care pune la dispoziția utilizatorului facilități suplimentare. Pentru a include acest driver, trebuie să răspundeți afirmativ la următoarea întrebare în timpul configurării nucleului:

Linux și plăcile de sunet

În comparație cu cele ale sistemelor Apple Macintosh, posibilitățile de generare a sunetului încorporate în calculatoarele compatibile IBM PC sunt relativ limitate. În esență, difuzorul încorporat al unui PC este capabil de redarea unei singure note. Nici măcar volumul sunetului respectiv nu poate fi controlat.

Posibilitățile unui calculator de redare a sunetelor pot fi îmbunătățite în mare măsură prin instalarea unei plăci de sunet, cum este Sound Blaster, care poate sintetiza o gamă variată de sunete. *Placa de sunet*, un adaptor care se conectează într-un slot de pe placa de bază a calculatorului, include circuitele electronice necesare pentru redarea și înregistrarea sunetelor. La o placă de sunet pot fi conectate difuzoare și un microfon. Multe plăci de sunet mai includ o interfață prin intermediul căreia puteți conecta o interfață CD-ROM.

Când se conectează un microfon la placa de sunet, acesta poate converti unde sonore *analogice* (care variază în mod continuu) în numere pe 8 biți sau 16 biți, eșantionând unda la frecvențe cuprinse între 4 kHz și 44 kHz (44000 de ori pe secundă). Frecvențe de eșantionare mai mari și un număr mai mare de biți (16) asigură o mai bună calitate, dar aveți nevoie de un spațiu pe disc mai mare pentru a stoca sunete de calitate mai bună. De asemenea, placa de sunet poate converti eșantioanele digitale de sunet în semnale analogice care pot fi redade pe un difuzor.

Majoritatea plăcilor de sunet, inclusiv populara Sound Blaster, conțin suport și pentru comenzile MIDI în afara înregistrării și redării sunetelor de tip waveform. MIDI (abrevierea de la *Musical Instrument Digital Interface* – interfață digitală cu instrumente muzicale) este frecvent folosit pentru înregistrarea și redarea sunetelor muzicale care pot fi create de un sintetizator. (Majoritatea plăcilor de sunet dispun de sintetizatoare încorporate.)

În continuare este prezentată o listă de plăci de sunet suportate de driverul de sunet Linux:

- 6850 UART MIDI
- Adlib (OPL2)
- plăcile ECHO-PSS (Orchid Soundwave32, Cardinal DSP16)
- Ensoniq SoundScape
- Gravis Ultrasound
- Gravis Ultrasound ACE
- placa anexă Gravis Ultrasound cu eșantionare pe 16 biți
- Gravis Ultrasound Max
- Logitech SoundMan Games (SBPro, suport stereo pe 44 kHz)
- Logitech SoundMan Wave (Jazz16/OPL14)
- Logitech SoundMan 16 (compatibilă PAS-16)
- MPU-401 MIDI (Roland)
- Media TriX AudioTriX Pro
- Media Vision Premium 3D (Jazz16)
- Microsoft Windows Sound System (MSS/WSS)
- plăcile OAK OTI-601D (Mozart)
- Orchid SW32
- OPTi 82C928/82C929 (MAD16/MAD16 Pro)

- Personal Sound Systems (PSS)
- Pro Audio Spectrum 16
- Pro Audio Studio 16
- Pro Sonic 16
- Sound Blaster 1.0
- Sound Blaster 16
- Sound Blaster 16ASP
- Sound Blaster 2.0
- Sound Blaster AWE32
- Sound Blaster Pro
- sunetul sistemului notebook Texas Instruments TM4000M
- ThunderBoard
- Turtle Beach Tropez
- Turtle Beach Maui
- sintetizatoarele Yamaha FM (OPL2, OPL3 și OPL4)



Secret

Multe plăci de sunet includ o interfață încorporată prin intermediul căreia puteți atașa o unitate CD-ROM. Linux necesită un driver de periferic încorporat pentru a accesa o unitate CD-ROM conectată la o placă de sunet; driverul de sunet nu are nici o legătură cu unitatea CD-ROM, chiar dacă unitatea este atașată la placa de sunet. Driverul de sunet nu are nici o legătură nici cu portul de joystick inclus în multe plăci de sunet.

Instalarea plăcii de sunet

Linux necesită un driver pentru controlul plăcii de sunet. Nucleul Linux inițial (după instalarea sistemului Linux de pe CD-ROM-ul anexat cărții) nu include driverul de sunet. Directorul `/usr/src/linux/drivers/sound` conține codul sursă pentru driverul de sunet.



Referință încrucișată

Capitolul 2 vă prezintă procesul de regenerare a nucleului Linux. Așa cum se vede în etapele prezentate, când configurați nucleul cu comanda `make config`, sunteți întrebati dacă doriți să adăugați suportul pentru placa de sunet. Dacă răspundeți afirmativ, procesul de configurare vă solicită informații suplimentare despre placa de sunet.

Mai întâi, programul de configurare dorește să afle dacă intenționați să includeți suport pentru anumite plăci de sunet. Trebuie să răspundeți afirmativ numai la întrebarea referitoare la marca plăcii de sunet inclusă în sistemul dumneavoastră. După ce ați specificat o anumită placă de sunet, programul de configurare va omite acele plăci care nu pot coexista cu placa selectată.

După ce selectați plăcile de sunet pe care doriți ca driverul de sunet să le suporte, programul de configurare solicită anumiți parametri pentru fiecare placă selectată. Trebuie să furnizați adresa portului I/O, IRQ și numărul canalului DMA. Trebuie să furnizați pentru acești parametri aceleași valori cu care au fost setați în DOS sau Windows. Trebuie să găsiți aceste setări prin lectura manualului plăcii de sunet (un calculator nou cu o placă de sunet include și un mic manual al plăcii).

Sistemul meu are o placă Sound Blaster. Sub MS-DOS, fișierul AUTOEXEC.BAT include următoarea comandă:

```
SET BLASTER=A220 I5 D1 H5 P300 T6
```

Din experiența mea cu plăcile Sound Blaster, am putut determina următoarele date din comanda SET: adresa I/O este 220 (în hexazecimal), IRQ este 5 și canalul DMA este 1. Astfel, am putut furniza aceste valori când am configurat nucleul Linux pentru suportul plăcii de sunet.

Configurarea driverului de sunet

În timpul configurării driverului de sunet, trebuie să răspundeți la mai multe întrebări puse de programul de configurare. Această secțiune pune la dispoziția utilizatorului unele îndrumări pentru răspunsul la anumite întrebări. Observați că programul de configurare nu pune fiecare întrebare, deoarece succesiunea exactă a întrebărilor depinde de răspunsurile dumneavoastră anterioare.

Pro AudioSpectrum 16 support? – Trebuie să răspundeți afirmativ (y) numai dacă dispuneți de o placă Pro AudioSpectrum 16 (desemnată sub numele PAS16), Pro Audio Studio 16 sau Logitech SoundMan 16. În particular, nu răspundeți afirmativ dacă dispuneți de o altă placă de sunet Media Vision sau Logitech, deoarece aceste plăci nu sunt compatibile PAS-16.

SoundBlaster support? – Răspundeți afirmativ (y) dacă placa de sunet a calculatorului dumneavoastră are marca Creative Labs Sound Blaster sau una dintre numeroasele clone compatibile hardware ale acesteia, cum ar fi Logitech sau Logitech SoundMan Games. Mai puteți răspunde afirmativ dacă placa de sunet se declară compatibilă Sound Blaster.

Generic OPL2/OPL3 FM synthesizer support? – Răspundeți afirmativ dacă placa dumneavoastră de sunet dispune de un chip FM produs de Yamaha (OPL2/OPL3/OPL4). Răspunsul afirmativ este o opțiune fără riscuri.

Gravis Ultrasound support? – Răspundeți afirmativ dacă placa de sunet a sistemului dumneavoastră este un Gravis Ultrasound (cu un nume ca GUS sau GUS MAX). Deși puteți include suport GUS fără conflicte cu alte plăci, driverul GUS consumă un mare volum de memorie. Ca atare, trebuie să răspundeți negativ (n) dacă nu dispuneți de GUS.

MPU-401 support (NOT for SB16)? – Deși interfața MIDI Roland MPU401 este suportată de multe plăci de sunet, modalitatea exactă de manipulare a acestei interfețe variază de la o placă la alta. Trebuie să răspundeți negativ (n) dacă placa dumneavoastră de sunet nu dispune de o interfață MPU401. Se recomandă răspunsul afirmativ dacă placa dumneavoastră de sunet dispune într-adevăr de o interfață MIDI 401 veritabilă.

6850 UART MIDI support? – Probabil că puteți răspunde negativ fără nici un fel de riscuri, deoarece interfața UART 6850 este rareori folosită. UART este abrevierea de la *Universal Asynchronous Receiver Transmitter* (receptor-transmițător universal asincron), circuit care împarte un bait în biți individuali și care recompune biții în baiți.

PSS (ECHO-ADI2111) support? – Prin răspunsul afirmativ la această întrebare se va include suport pentru plăcile de sunet bazate pe cipsetul PSS. Trebuie să răspundeți afirmativ numai dacă aveți un Orchid SW32, Cardinal DSP16 sau o altă placă de sunet bazată pe cipsetul PSS.

16-bit sampling option of GUS (not GUS MAX)? – Placa Gravis Ultrasound dispune de o placă anexă pe care se poate instala o placă de eșantionare pe 16 biți. Trebuie să răspundeți negativ dacă aveți o placă GUS MAX. Dacă răspundeți afirmativ, suportul pentru GUS MAX este dezactivat în mod automat.

GUS MAX support? – Dacă aveți o placă GUS MAX, răspundeți afirmativ la această întrebare.

Microsoft Sound System support? – Trebuie să răspundeți afirmativ dacă dispuneți de placa originală Windows Sound System produsă de Microsoft sau Aztech SG 16 Pro (sau NX 16 Pro).

Ensoniq SoundScape support? – Răspundeți afirmativ dacă placa dumneavoastră de sunet este bazată pe cipsetul Ensoniq SoundScape. Mai mulți producători – precum Ensoniq, Spea și Reveal (care mai produce și alte tipuri de plăci) – folosesc acest cipset.

Sound Blaster Pro support? – Răspundeți afirmativ dacă placa dumneavoastră de sunet este un Sound Blaster Pro, Sound Blaster 16 sau o placă compatibilă.

Sound Blaster 16 support? – Răspundeți afirmativ pentru a activa acest suport dacă sistemul dumneavoastră dispune de o placă de sunet Sound Blaster 16.

/dev/dsp and /dev/audio support (usually required)? – Răspundeți afirmativ pentru ca programul de configurare să creeze în mod automat punctele de montare: /dev/dsp și /dev/audio.

MIDI interface support? – Răspundeți afirmativ pentru a activa perifericele MIDI (care au nume de genul /dev/midi*NN*, unde *NN* este un număr) și permite acces la placa de sunet prin intermediul perifericelor /dev/sequencer și /dev/music. Un răspuns afirmativ are efect asupra tuturor perifericelor compatibile MPU401 și/sau MIDI.

FM synthesizer (YM3812/OPL-3) support? – Puteți răspunde fără probleme afirmativ la această întrebare.

/dev/sequencer support? – Dacă răspundeți negativ, programul de configurare dezactivează dispozitivele /dev/sequencer și /dev/music.

Numele perifericelor de sunet

Ca și orice alt periferic în Linux, perifericele de sunet au fișiere în directorul /dev, cu nume specifice. Tabelul 12.3 prezintă numele de fișiere de dispozitiv standard care furnizează facilități de sunet în Linux (este posibil ca nu toate aceste fișiere să existe în sistemul dumneavoastră).

Tabelul 12.3 Nume de fișiere standard de periferice de sunet în Linux

Nume fișier de periferic	Descriere
/dev/audio	Un dispozitiv audio capabil de a reda fișiere audio compatibile cu stațiile de lucru Sun (de regulă cu o extensie .au). Dispozitivul nu poate suporta toate caracteristicile de sunet ale unui periferic audio inclus într-o stație de lucru Sun, dar poate reda fișiere audio Sun.
/dev/audio1	Periferic audio compatibil cu stațiile de lucru Sun pentru a doua placă de sunet (dacă există).
/dev/dsp	Periferic de procesare a semnalelor digitale, de asemenea capabil de redarea fișierelor audio Sun.
/dev/midi	Periferic MIDI.
/dev/mixer	Consolă de mixaj de sunet.
/dev/mixer1	A doua consolă de mixaj de sunet.

(continuare)

Tabelul 12.3 Continuare

Nume fișier de periferic	Descriere
<code>/dev/pcaudio</code>	Echivalent cu <code>/dev/audio</code> , dar folosește difuzorul calculatorului. (Trebuie să instalezi un driver separat pentru difuzorul calculatorului, disponibil la adresa ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/ .)
<code>/dev/pcmixer</code>	Echivalent cu <code>/dev/mixer</code> , dar folosește difuzorul calculatorului. (Trebuie să instalezi un driver separat pentru difuzorul calculatorului, disponibil la adresa ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/ .)
<code>/dev/sequencer</code>	Ordonator MIDI.
<code>/dev/sndstat</code>	Periferic care asigură informații despre placa de sunet (vezi exemplul din paragraful „Verificarea stării driverului de sunet”).

Testarea plăcii de sunet

După ce activați suportul pentru placa dumneavoastră de sunet și regenerați nucleul Linux, trebuie să reîncărcați sistemul de operare (intrați în sistem ca root și folosiți comanda `/sbin/shutdown -r now`). La încărcarea sistemului de operare, veți vedea un mesaj referitor la driverul de sunet.



Sugestie

Dacă mesajele de boot se derulează prea rapid pentru a fi văzute, tastezi comanda `dmesg | more` pentru a vedea aceste mesaje pe ecran unul câte unul. Căutați una sau mai multe linii care încep cu `snd`. Pe sistemul meu, de exemplu, se află instalată o placă de sunet Sound Blaster Pro. Mesajele de boot includ următoarea linie:

```
snd2 <SoundBlaster Pro 4.13> at 0x220 irq 5 drq 1
```

În acest caz, textul încadrat între parantezele unghiulare afișează numele plăcii de sunet și numărul de versiune al circuitelor de procesare a semnalelor digitale de pe placa de bază. Mesajul de boot referitor la perifericul de sunet afișează adresa I/O 0x220 (220 în notație hexazecimală), un IRO egal cu 5 și un canal DMA (denumit *drq*) de 1.

Uneori, driverul de sunet poate afișa mesaje de eroare sau de avertisment la rularea secvenței de boot a sistemului. Când rulați comanda `dmesg`, trebuie să fiți atent și la asemenea mesaje.

Verificarea stării driverului de sunet

Dacă observați un mesaj de boot referitor la placa de sunet, puteți vedea că suportul pentru sunet este inclus în nucleu. Driverul de sunet vine cu o altă modalitate de verificare a stării.



Secret

```
cat /dev/sndstat
Sound Driver:3.5.4-960630
Kernel: Linux localhost 2.0.32 #1 Fri Jan 30 07:08:01 EST 1998 i586
Config options: 0
```

Installed drivers:
Type 2: Sound Blaster

```
Card config:
Sound Blaster at 0x220 irq 5 drq 1
OPL-2/OPL-3 FM at 0x338 drq 0
```

Audio devices:
0: Sound Blaster Pro (4.13)

Synth devices:
0: Yamaha OPL-3

Midi devices:

Timers:
0: System clock

Mixers:
0: Sound Blaster

Mesajele apărute pe sistemul dumneavoastră vor fi diferite, deoarece probabil că dispuneți de o placă de sunet cu alți parametri decât a mea. Cu toate acestea, fișierul de periferic /dev/sndstat oferă informații de diagnostic performante despre driverul de sunet.

Testarea plăcii de sunet

Pentru a testa driverul de sunet, încercați să rulați un fișier de sunet. Tot ce aveți de făcut este să trimiteți fișierul de sunet către perifericul de sunet (/dev/audio) cu comanda cat.

Dispozitivul de sunet poate reda fișierele de sunet Sun, al căror nume se încheie cu extensia .au. Pentru a reda un fișier de sunet denumit piano-beep.au, veți lansa următoarea comandă:

```
cat piano-beep.au > /dev/audio
```



Sugestie

Dacă aveți montat CD-ROM-ul Linux Red Hat, veți descoperi numeroase fișiere .au în directorul `/mnt/cdrom/live/usr/lib/games/xboing/sounds`. Încercați următoarea comandă:

```
cat /mnt/cdrom/live/usr/lib/games/xboing/sounds/evillaugh.au > /dev/audio
```

Dacă fisierul de sunet se aude, atunci placa de sunet funcționează corect.

Dacă aveți un microfon conectat la placa de sunet, mai puteți încerca să înregistrați un fișier de sunet cu durată de 10 secunde cu următoarea comandă:

```
dd bs=8k count=10 </dev/audio >test.au
```

Comanda dd pur și simplu copiază o cantitate determinată de date dintr-un fișier în altul. În acest caz, fișierul de intrare este perifericul audio (care înregistrează de la microfon), iar fișierul de ieșire este fișierul de sunet. După ce ați înregistrat fișierul de sunet, acesta poate fi rulat prin trimiterea datelor înapoi la /dev/audio folosind comanda cat.



Pentru a folosi placa de sunet în mod adecvat, aveți nevoie de unele aplicații care vă permit să înregistrați și să redați fișiere de sunet. Încercați una dintre următoarele resurse Internet pentru a localiza aplicații de înregistrare și redare a sunetelor pentru Linux:

- <ftp://sunsite.unc.edu:/pub/Linux/apps/sound/>
- <ftp://tsx-11.mit.edu:/pub/linux/packages/sound/>
- <ftp://nic.funet.fi:/pub/Linux/util/sound/>

Diagnosticarea plăcilor de sunet

După ce ați configurat driverul de sunet și ați executat reboot, dacă placa de sunet nu generează nici un sunet când copiați fișiere audio în fișierul /dev/audio, încercați următoarea succesiune de etape pentru a determina și remedia problema:

1. Verificați dacă folosiți nucleul pe care l-ați regenerat cu suport pentru placa de sunet. Pentru a vedea numărul de versiune, folosiți comanda `uname -a`. În continuare este prezentată o ieșire specifică a comenzii `uname -a`:

```
Linux localhost 2.0.32 #1 Fri Jan 30 07:08:01 EST 1998 i586 unknown
```



Referință
încrucișată

Acest șir prezintă numărul de versiune al nucleului, ca și data la care a fost generat acesta. Dacă această dată nu corespunde datei la care ați regenerat nucleul, înseamnă că nu rulați noul nucleu. Parcurgeți etapele descrise în capitolul 2 și verificați dacă într-adevăr ați instalat noul nucleu. O altă problemă comună este când uităm să reîncărcăm sistemul de operare, deci încercați și aceasta.

2. Verificați dacă driverul de sunet este inclus în nucleu. O modalitate de a verifica acest lucru o constituie examinarea conținutului fișierului /proc/devices. Un exemplu este prezentat în continuare:

```
cat /proc/devices
Character devices:
```

```
1 mem
2 pty
3 tty
4 ttty
5 cua
7 vcs
10 misc
```

```
Block devices:
```

```
1 ramdisk
2 fd
3 ide0
7 loop
8 sd
9 md
22 ide1
```

Listul trebuie să conțină dispozitivul de tip caracter 14, denumit sound. Dacă nu vedeți acest periferic, înseamnă că nu ați configurat nucleul în mod corect pentru a include driverul de sunet. Reconfigurați și regenerați nucleul, asigurându-vă că ați inclus suport pentru placa dumneavoastră de sunet.

3. Verificați dacă driverul de sunet a detectat placa de sunet la lansarea sistemului. Folosiți comanda `dmesg | more` pentru a examina mesajele de boot și a detecta o linie care semnalează că placa de sunet a fost găsită. Dacă driverul de sunet nu semnalează placa de sunet, este posibil ca aceasta să nu fie corect instalată și/sau configurată.

Mai întâi, verificați dacă placa de sunet funcționează sub DOS sau Windows. Determinați adresa I/O, IRQ și canalul DMA, după care reconfigurați nucleul pentru a include suport pentru placa dumneavoastră de sunet cu aceiași parametri ca sub DOS.

4. Dacă nimic nu pare să aibă efect, citiți ultimul document Sound-HOWTO. Pentru a citi documentele HOWTO, tastați `cd /usr/doc/HOWTO` pentru a comuta în directorul unde se află localizate fișierele HOWTO. Apoi tastați `zcat Sound-HOWTO.gz | more` pentru a vizualiza fișierul Sound-HOWTO.



Referință
încrucișată

Dacă nici acum nu reușiți să „convingeți” placa de sunet să funcționeze sub Linux, inserați o notiță într-unul din grupurile de discuții comp.os.linux. În capitolul 19 se discută despre modul de conectare la Internet și de accesare a grupurilor de lucru.

Rezolvarea unor probleme comune legate de placa de sunet

În afară de configurarea elementară a driverului pentru placa de sunet, puteți avea probleme cu anumite plăci de sunet. Această secțiune conține sugestii pentru rezolvarea problemelor întâlnite mai frecvent.

Funcționează sub DOS, nu și sub Linux

Multe plăci de sunet compatibile Sound Blaster sunt funcționale sub MS-DOS și Microsoft Windows, dar nu funcționează sub Linux. Un simptom specific este următorul: totul merge perfect – driverul de sunet este corect instalat și raportează placa de sunet cu parametri corecți la inițializare – dar nu auziți nici un sunet când copiați un fișier de sunet în fișierul /dev/audio. În acest caz, trebuie mai întâi să inițializați placa de sunet sub DOS și Windows. Încercați următoarele etape:

1. Inițializați sistemul sub DOS și rulați Windows 3.1. Dacă aveți Windows 95, lansați Windows 95.
2. Rulați aplicația Media Player și derulați câteva fișiere de sunet sub Windows.
3. Părăsiți Windows și reinițializați calculatorul. Dacă rulați Windows 3.1, părăsiți Windows și apăsați Ctrl-Alt-Delete sub DOS. În Windows 95, selectați opțiunea Shut Down a meniului Start și selectați opțiunea Reboot din caseta de dialog rezultantă.
4. Asigurați-vă că executați reboot din Linux. Dacă folosiți LILO, apăsați Shift și răspundeți la prompt-ul LILO. Dacă folosiți o dischetă de boot Linux, introduceți discheta în unitatea A înainte de a reexecuta boot.

După încărcarea sistemului de operare Linux, driverul de sunet ar trebui să deruleze fără probleme fișierul de sunet.

Poate reda sunet, dar nu poate înregistra

Dacă placa de sunet poate reda sunete, dar nu poate înregistra, mai întâi asigurați-vă că puteți înregistra sunete sub DOS și Windows. Acest lucru arată că placa de sunet este capabilă de a executa înregistrări.

Apoi procurați-vă un program de mixaj sonor (dintr-unul din site-urile FTP prezentate în paragraful „Testarea plăcii de sunet”) și folosiți-l pentru a selecta dispozitivul de înregistrare corespunzător, cum este microfonul. De asemenea, puteți regla amplificarea la intrare (input gain) pentru a vedea dacă are vreun efect.

Adaptorul Pro Audio Spectrum PAS16 și adaptorul SCSI Adaptec 1542

Placa de sunet PAS16 folosește adresa I/O 330H (330 în hexazecimal) ca adresă de port MIDI. Placa SCSI Adaptec AHA 1542 poate folosi, de asemenea, adresa I/O 330H. Cu toate acestea, Linux poate folosi placa AHA 1542 cu adresa I/O de bază de 334H. Pentru a evita conflictul de adrese I/O între PAS16 și placa SCSI AHA 1542, setați adresa I/O a plăcii SCSI la 334H.



Secret

O altă problemă o constituie conflictele DMA între PAS16 și AHA 1542, care pot bloca sistemul datorită unei erori de memorie. Soluția este de a modifica doi parametri – timpii BUS ON și BUS OFF – ai plăcii SCSI AHA 1542. Procurați-vă programul SCSEL.EXE de la Adaptec prin Internet, rulați-l sub DOS și reduceți timpul BUS ON sau măriți timpul BUS OFF până la dispariția problemei.

Placa Sound Blaster AWE32 nu este suportată

Linux nu include suport pentru sintetizatorul MIDI E-mu și cipul ASP pe plăcile de sunet Sound Blaster AWE32. Problema este că firma Creative Labs, producătorul plăcii AWE32, nu a făcut publice detaliile de programare privind plăcile E-mu și ASP.

Rezumat

Spre deosebire de multe stații de lucru UNIX, un PC obișnuit conține o unitate CD-ROM și o placă de sunet. Popularitatea programelor multimedia alimentează cererea de unități CD-ROM și de posibilități de sunet pentru calculatoare personale. De asemenea, multe pachete software, inclusiv Linux, sunt incluse pe CD-ROM. Linux include suport pentru unități CD-ROM și plăci de sunet. Acest capitol descrie unitățile CD-ROM și plăcile de sunet suportate de Linux și vă arată cum să detectați și să corectați unele dintre problemele de instalare ce apar la plăcile de sunet și la unitățile CD-ROM.

Din lectura acestui capitol, ați învățat următoarele:

- ▶ Suportul Linux pentru o anumită unitate CD-ROM depinde de interfața folosită pentru a conecta unitatea CD-ROM respectivă la placa de bază a sistemului. Linux include suport pentru cele mai populare interfețe ATAPI (IDE) și SCSI pentru unitățile Linux, precum și pentru multe interfețe CD-ROM patentate, cum sunt Aztech, Sony, Mitsumi și interfața CD-ROM Sound Blaster Pro.
- ▶ Utilizatorii începători selectează frecvent, din greșeală, un driver CD-ROM bazat pe numele producătorului unității CD-ROM. În schimb, trebuie selectat un driver CD-ROM care corespunde interfeței unității. Astfel, o unitate CD-ROM Mitsumi cu o interfață ATAPI necesită driverul IDE, nu Mitsumi.
- ▶ Plăcile de sunet includ frecvent interfețe pentru unitățile CD-ROM și pentru joystick-uri. Placa de sunet suportă numai caracteristicile de sunet ale plăcii de sunet. Pentru unitatea CD-ROM și joystick aveți nevoie de drivere separate.
- ▶ Pentru a folosi o placă de sunet, trebuie să regenerați nucleul Linux. Când configurați nucleul, trebuie să activați suportul pentru placa de sunet și să furnizați informații despre adresa I/O, IRQ și DMA ale plăcii.
- ▶ Puteți folosi perifericele `/dev/audio` și `/dev/sndstat` pentru a verifica placa de sunet după regenerarea nucleului și reinițializarea sistemului cu noul nucleu.

Capitolul 13 Tastaturi și dispozitive de indicare

În acest capitol:

- ▶ Veți ajusta intervalul de repetare automată al tastaturii și viteza de repetare în Linux
- ▶ Veți înțelege conceptul de hartă a tastaturii
- ▶ Veți configura tastatura pentru unele limbi europene
- ▶ Veți examina tipurile de interfață cu mouse-ul și protocoale de mouse suportate de Linux
- ▶ Veți configura XFree86 pentru a utiliza diferite tipuri de tastaturi și mouse

Tastatura și mouse-ul (sau un alt dispozitiv de indicare similar cu mouse-ul, denumit *trackball*) reprezintă mecanismele elementare de furnizare a datelor de intrare pentru calculator. În limitele compatibilității hardware, Linux acceptă orice tastatură care funcționează sub DOS. Cu toate acestea, uneori este necesară modificarea anumitor caracteristici ale tastaturii, cum ar fi viteza de repetare a unui caracter când se ține apăsată o tastă. De asemenea, se mai poate asocia un alt set de caractere cu tastele fizice, mai ales dacă se folosește o altă limbă decât engleza. Acest capitol descrie facilitățile de personalizare a tastaturii oferite de Linux.

Nu aveți nevoie de un mouse (eu folosesc termenul de *mouse* pentru a desemna orice dispozitiv de indicare, inclusiv un trackball) pentru a rula Linux. Dacă, totuși, instalați și utilizați XFree86 (X Window System pentru Linux) veți avea nevoie de un mouse. Există mai multe mărci de mouse și numeroase tipuri de interfețe (cum ar fi mouse-ul serial sau mouse-ul de magistrală). În acest capitol, veți învăța despre tipurile de interfață cu mouse-ul și despre protocoalele de mouse suportate de Linux și X.

Linux și tastaturile

Tastatura este una din acele anexe indispensabile ale unui calculator, dar pe care o luăm ca pe ceva de la sine înțeles și căruia nu-i acordăm prea multă atenție. Dacă o tastatură este funcțională sub DOS, atunci ar trebui să fie compatibilă cu Linux. Astfel, producătorul și modelul tastaturii nu prea reprezintă o problemă.

Desigur, pot fi modificate unele caracteristici fizice ale tastaturii, precum următoarele:

- *Intervalul de repetare automată:* dacă țineți apăsată o tastă, tastatura așteaptă un anumit interval de timp înainte de a începe să repete afișarea tastei respective.
- *Viteza de repetare.* Viteza de repetare este viteza cu care tastatura repetă afișarea unei taste.

Un alt aspect care poate fi adaptat are legătură mai mult cu Linux decât cu tastatura în sine. Acest aspect determină modul în care Linux interpretează apăsarea unei taste. Puteți determina tasta Backspace să genereze combinația Ctrl-H prin *remaparea* tastei, adică prin schimbarea modului prin care o apăsare de tastă este asociată unei anumite acțiuni.

În secțiunile următoare vom prezenta ambele tipuri de personalizare a tastaturii: caracteristicile fizice și maparea tastelor pe diferite acțiuni.

Terminologie și notații folosite pentru tastaturi

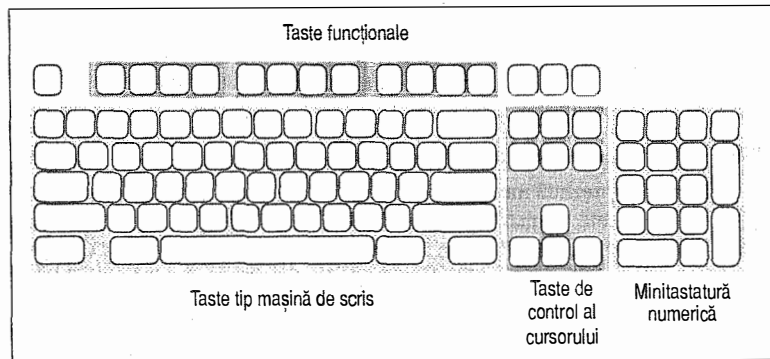
În acest capitol, veți întâlni referințe la anumite taste (sau combinații de taste) de la tastatură. Această secțiune reprezintă o scurtă trecere în revistă a numelor tastelor și a altor termeni asociați tastaturilor.

Macheta tastaturii

Noțiunea de *machetă a tastaturii* se referă la modalitatea de dispunere a tastelor pe tastatură. Majoritatea calculatoarelor folosesc o tastatură cu 101 taste. Figura 13.1 prezintă o machetă tipică pentru o tastatură de PC cu 101 taste.

Figura 13-1

Macheta unei tastaturi pentru PC cu 101 taste



Tastele alfabetice de bază și tastele funcționale ocupă majoritatea spațiului de pe tastatură. Un set de 12 taste funcționale – numerotate de la F1 la F12 – apare de-a lungul laturii superioare a tastaturii. Minitastatura numerică reprezintă blocul de taste din partea dreaptă a tastaturii.

Colțul din dreapta sus al tastaturii dispune de trei lumini de stare, denumite LED-uri (*light emitting diode* – diodă electroluminiscentă, o componentă electronică emițătoare de lumină). LED-urile – denumite Num Lock, Caps Lock și Scroll Lock – arată care dintre tastele Lock corespunzătoare este apăsată.



Observație

Tastele Ctrl, Shift și Alt sunt cunoscute sub numele de *taste modificador*. După cum știți, tastatura dispune de câte o pereche de asemenea taste. În documentația Linux, tastele Alt din partea dreaptă sunt denumite *taste AltGr*.

Deși multe dintre tastele de la tastatură sunt etichetate similar (de exemplu, numerele de pe minitastatura numerică sunt identice cu cele de pe tastele situate în partea superioară a blocului principal de taste), Linux atribuie un cod cheie unic fiecărei taste de pe tastatură. De regulă, aceste coduri sunt cuprinse între 1 și numărul tastelor de pe tastatură.

Terminologia aferentă tastaturii în X

Sistemul X Window System (pe scurt X) atribuie de asemenea un cod cheie fiecărei taste de pe tastatură. Din păcate, codurile cheie X nu sunt identice cu cele folosite de Linux. Intern, X convertește codurile cheie fizice în *simboluri de tastă* (*keysyms*), care reprezintă nume simbolice ale tastaturilor.

Taste precum Shift, Alt, Ctrl sau Caps Lock sunt cunoscute sub numele de *taste modificador*, deoarece modifică semnificațiile altor taste. X suportă maximum cinci taste modificador dependente de sistem.

Întârzierea de repetare și viteza de repetare

La încărcarea sistemului de operare Linux, nucleul setează viteza de repetare a tastei la valoarea maximă admisă de tastatură. Deși această viteză maximă este admisă de majoritatea tastaturilor, în unele situații un caracter se repetă mult prea ușor, deoarece viteza de repetare este mult prea mare. Într-un asemenea caz, puteți folosi programul *kbrdate* pentru a reduce viteza de repetare a tastaturii.

Programul utilitar *kbrdate* vă permite să manipulați viteza de repetare a tastaturii și intervalul de întârziere, în speță intervalul de timp în care o tastă trebuie apăsată înainte de a se începe repetarea afișării caracterului respectiv.

În continuare este prezentată sintaxa programului *kbrdate*:

```
kbrdate [-s] [/r rate] [-d delay]
```

Toate argumentele incluse între paranteze sunt opționale. Dacă rulați *kbrdate* fără opțiuni, rezultatul va fi următorul:

kbrdate

```
Typematic Rate set to 10.9 cps (delay = 250 ms)
```

Acest mesaj de la *kbrdate* semnifică faptul că viteza de repetare a tastaturii (denumită de IBM *typematic rate* la prima utilizare a acestui tip de tastatură) este setată la 10,9 caractere pe secundă (cps), iar intervalul de repetare este 250 milisecunde (ms). Astfel, când programul rulează fără opțiuni, *kbrdate* setează viteza de repetare și întârzierea la aceste valori prestabilite.

Viteza poate fi stabilită folosind opțiunea *-r* și poate fi cuprinsă între 2 și 30 cps. De regulă, tastatura permite un set de valori discrete. Un set comun de astfel de valori este: 2,0; 2,1; 2,3; 2,5; 2,7; 3,0; 3,3; 3,7; 4,0; 4,3; 4,6; 5,0; 5,5; 6,0; 6,7; 7,5; 8,0; 8,6; 9,2; 10,0; 10,9; 12,0; 13,3; 15,0; 16,0; 17,1; 18,5; 20,0; 21,8; 24,0; 26,7 și 30,0 cps. Când specificați o valoare, programul *kbrdate* selectează din acest set o valoare mai mică sau egală cu cea solicitată. Astfel, dacă introduceți comanda *kbrdate -r 18.0*, programul va selecta valoarea 17,1.

Pentru a stabili valoarea întârzierii folosiți opțiunea *-d*, care poate fi cuprinsă între 250 și 1000 ms (sau o secundă). Sunt permise numai patru valori posibile: 250, 500, 750 și 1000 ms.

Opțiunea *-s* determină rularea programului *kbrdate* fără afișare de mesaje.

Programul *kbrdate* nu funcționează cu anumite tastaturi. Se pare că programul este incapabil de a seta viteza de repetare pe tastatura Gateway AnyKey.



Sugestie

Când rulați sistemul X Window System, acesta preia controlul asupra tastaturii. Sub X, puteți rula *kbrdate* (dintr-o fereastră *xterm*) și puteți seta valorile vitezei de repetare și ale întârzierii. De asemenea, X mai include utilitarul *xset*, care vă permite să modificați diferite preferințe de utilizator pentru tastatură, mouse și monitor. Deși *xset* nu vă permite să stabiliți viteza de repetare, puteți activa sau dezactiva facilitatea de repetare. Pentru a dezactiva această funcție, tasteți următoarea comandă la prompt-ul interpretorului într-o fereastră *xterm*:

```
xset r off
```

Încercați să mențineți apăsată o tastă. Repetiția caracterului nu se va produce. Pentru a activa facilitatea de repetare automată, folosiți următoarea comandă:

```
xset r on
```

Harta tastaturii

Probabil că utilizatorul vorbitor de limbă engleză nu acordă prea multă atenție modului în care tastatura mapează tastele fizice pe caractere, deoarece pentru limba engleză există o corespondență unu-la-unu între tastele fizice și literele alfabetului.

Situația nu este la fel de simplă dacă limba dumneavoastră nativă nu este engleza și aveți nevoie de unele caractere speciale, cum sunt cele cu *umlaut* (două puncte deasupra literei) sau de alte litere cu accent. Pentru a introduce asemenea caractere de la tastatură, trebuie să schimbați modul de conversie al apăsării unei taste în caractere (în Linux). Conversia respectivă poartă numele de *hartă a tastaturii* sau *conversia tastaturii*.

Rețineți că introducerea unor litere speciale de la tastatură necesită două etape. Mai întâi, trebuie să determinați Linux să interpreteze apăsarea unei taste drept caracterul special dorit. Apoi, aveți nevoie de un program capabil să afișeze caracterul special. Modul de afișare a caracterului respectiv depinde de fontul folosit.



Observație

Linux include două programe – `loadkeys` și `xmodmap` – pentru modificarea felului în care apăsarea unei taste fizice este asociată unei anumite acțiuni. Programul folosit depinde de modul de rulare: `mod text` sau `X Window System`. Când lucrați într-o consolă virtuală Linux (în esență ecranul text), apăsările pe taste sunt manipulate de driverul de tastatură. În acest caz, pentru a schimba modul de conversie a tastelor se va folosi programul `loadkeys`.

Când X rulează, acesta preia controlul atât asupra tastaturii, cât și a mouse-ului. Ca atare, va trebui să folosiți un alt program utilitar, `xmodmap`, pentru a modifica harta tastaturii. Totuși, X consultă la inițializare tabelele de conversie a tastelor din Linux, deci există o oarecare consecvență între conversiile de tastatură efectuate în `mod text` și în X.

În Linux, driverul de tastatură consultă o tabelă a tastaturii pentru a determina modul de conversie a unui cod de tastă fizic într-un șir de alte coduri sau într-o acțiune. O asemenea tabelă de conversie a tastaturii (sau hartă a tastaturii) poate fi încărcată folosind programul `loadkeys`.

Harta tastaturii este definită într-un fișier text stocat în format comprimat. Harta prestabilită a tastaturii se găsește în fișierul `/usr/lib/kbd/keytables/defkeymap.map.gz`. Driverul de tastatură folosește acest tabel pentru a converti apăsările fizice pe taste în acțiuni.



Sugestie

Puteți găsi și alte fișiere tip hartă a tastaturii în directorul `/usr/lib/kbd/keytables`. Fiecare fișier reprezintă o hartă a tastaturii pentru diferite regiuni geografice sau țări. De exemplu, harta tastaturii pentru SUA se găsește în fișierul `us.map.gz`, iar harta Finlandei se află în fișierul `fi.map.gz` (fișierele sunt stocate în format comprimat, motiv pentru care numele fișierelor au extensia suplimentară `.gz`). Pentru a încărca o anumită hartă a tastaturii, folosiți programul `loadkeys` și specificați numele fișierului fără extensia `gz`, după cum urmează:

```
loadkeys /usr/lib/kbd/keytables/fi.map
```

După aceasta, dacă apăsați tasta minus, veți obține semnul plus – un simptom al modificării modului de asociere a tastelor fizice cu diferite caractere. Pentru a reveni la harta prestabilită a tastaturii, tastați `loadkeys -d` (ceea ce înseamnă încărcarea hărții prestabilite a tastaturii).

De fapt, puteți modifica harta tastaturii în mod interactiv, de la linia de comandă. Să considerăm un exemplu inutil, dar semnificativ. Să presupunem că doriți ca tasta A să fie

interpretată drept Q și invers. Aceasta înseamnă că, după remaparea tastelor, trebuie să apăsați pe tasta A pentru a afișa caracterul Q (și tasta Q pentru a introduce A). Înainte ca situația să se complice și mai mult, încercați să tastați următoarele la promptul interpretorului (procedura nu este funcțională sub X):

loadkeys

```
keycode 30 = +q
```

```
keycode 16 = +a
```

(apasati `Ctrl+D` pentru a indica sfarsitul datelor de intrare)

Acum observați rezultatul. Apăsați tastele A și Q de câteva ori și verificați dacă semnificațiile celor două taste s-au inversat între ele. Programul `loadkeys` așteaptă să introducăți linii de text. Pe fiecare linie veți specifica acțiunea corespunzătoare apăsării fizice a unei taste. Partea stângă a semnului egal (=) reprezintă codul tastei, iar partea dreaptă reprezintă acțiunea asociată codului respectiv. O literă cu semnul plus (+) înseamnă că codul cheie corespunde literei respective și că litera trebuie să comute din minusculă în majusculă și invers la apăsarea tastei Shift sau la activarea tastei Caps Lock.



Sugestie



Următor

Linux atribuie un număr – cod cheie – fiecărei taste de la tastatură. Puteți folosi comanda `showkey` pentru a determina codul cheie al oricărei taste de pe tastatură. Pur și simplu tastați `showkey` și apăsați tasta al cărei cod doriți să îl aflați.

Dacă configurați Linux pentru o hartă de tastatură neamericană, puteți inițializa harta tastaturii la pornirea sistemului. Tot ce aveți de făcut este să editați fișierul `/etc/sysconfig/keyboard`; acest fișier conține o singură linie, care specifică harta tastaturii care se va încărca la inițializarea sistemului. Pentru a încărca o hartă de tastatură americană, acest fișier conține următoarea linie:

```
KEYTABLE="/usr/lib/kbd/keytables/us.map"
```

Pentru a aplica tastaturii tabela de conversie pentru limba daneză, puteți folosi următoarea comandă:

```
/usr/bin/loadkeys /usr/lib/kbd/keytables/dk.map
```

Puteți vizualiza tabela curentă de conversie a tastaturii folosind programul `dumpkeys`. Dacă tastați `dumpkeys` fără opțiuni, programul afișează maparea de la coduri de tastă la numele tastelor și mai arată acțiunea care se petrece la apăsarea unei taste cu modificatori (cum sunt `Ctrl`, `Shift` și `Alt`).

Tastatura și XFree86

În XFree86, serverul X controlează tastatura și mouse-ul. Astfel, serverul X necesită unele informații despre tastatură. Aceste informații trebuie incluse în secțiunea `Keyboard` a fișierului `/etc/XF86Config`, în speță fișierul de configurare pentru XFree86. O secțiune `Keyboard` normală se prezintă astfel:

```
Section "Keyboard"
    Protocol "Standard"
    AutoRepeat      500    5
    LeftAlt         Meta
    RightAlt        ModeShift
EndSection
```

Tabelul 13.1 prezintă opțiunile care pot fi specificate în secțiunea `Keyboard`.

Tabelul 13.1 Opțiunile din secțiunea Keyboard a fișierului XF86Config

Opțiune	Funcție
Protocol "nume-protocol"	Specifică protocolul de tastatură. Pentru <i>nume-protocol</i> în Linux se poate folosi standard. Celălalt nume de protocol poate fi <i>xqueue</i> , adecvat pentru alte sisteme UNIX.
AutoRepeat rata de intarziere	Setează întârzierea de repetare și viteza de repetare (nu este funcțională pe toate sistemele). Întârzierea se măsoară în milisecunde, iar viteza este exprimată în caractere pe secundă.
Server NumLock	Determină serverul X să trateze intern tasta Num Lock. Această opțiune era necesară pentru versiunile X anterioare lui X11R6. În prezent nu mai este necesară specificarea acestei opțiuni, deoarece versiunea X de pe CD-ROM-ul anexat este X11R6.
LeftAlt tasta	Mapează tasta Alt din stânga la tasta specificată (vezi text pentru listă).
RightAlt tasta	Mapează tasta Alt din dreapta la tasta specificată (vezi text pentru listă).
AltGr tasta	Mapează tasta Alt din dreapta la tasta specificată (vezi text pentru listă).
ScrollLock tasta	Mapează tasta ScrollLock la tasta specificată (vezi text pentru listă).
RightCtrl tasta	Mapează tasta Ctrl din dreapta la tasta specificată (vezi text pentru listă).
XLeds x y z	Permite aplicațiilor X să preia controlul asupra luminilor de stare (LED-uri). Pentru valorile <i>x</i> , <i>y</i> și <i>z</i> se pot folosi valori cuprinse între 1 și 3.
VT SysReq	Determină serverul X să rezerve o secvență <i>Alt+SysRq+Fn</i> (<i>n</i> reprezintă un indicativ numeric de tastă funcțională) pentru comutarea terminalelor virtuale (VT). Dacă această linie este specificată, se dezactivează secvența de taste prestabilită de comutare la VT (<i>Ctrl-Alt-Fn</i>).
VT Init "comanda"	Determină serverul X să ruleze <i>comanda</i> specificată cu interpretorul (<i>/bin/sh -c</i>) după ce serverul X își deschide terminalul virtual.

Pentru a specifica maparea tastelor pentru tastele Alt stânga și dreapta, pentru tasta Scroll Lock și pentru tasta Ctrl din dreapta se poate folosi unul din următoarele nume de tastă (acestea sunt numele recunoscute de serverul X):

- Compose
- Control
- Meta
- ModeLock
- ModeShift
- ScrollLock



Observație

Acestea nu sunt decât numele standard ale tastelor în X. Fiecare tastă standard este asociată cu o tastă fizică de pe tastatură. De exemplu, ceea ce serverul X denumeste *tastă Meta* este tasta Alt din partea stângă a tastaturii.

În continuare este prezentată maparea prestabilită pentru tastele Alt stânga și dreapta, tasta ScrollLock și tasta Ctrl din dreapta:

- Alt stânga: Meta

- Alt dreapta (AltGr este un sinonim pentru Alt dreapta): Meta
- Ctrl dreapta: Control
- ScrollLock: Compose

În X, serverul X preia controlul asupra ecranului, a tastaturii și a mouse-ului. Avantajul este că X are propria sa schemă de mapare a apăsărilor fizice pe taste la diferite acțiuni. Din fericire, în Linux XFree86 folosește harta de tastatură Linux pentru a-și inițializa propria hartă de tastatură, asigurând astfel o oarecare consecvență între cele două hărți.



Observație

Serverul X este cel care stabilește codul tastă care va fi generat pentru o anumită tastă fizică. Fiecare tastă, inclusiv modificatorii, are un cod tastă unic. Deși codul tastă generat pentru tastele alfanumerice comune poate fi identic pentru multe stații de lucru, acest fapt nu este absolut sigur. Ca atare, aplicațiile X nu folosesc codul tastă brut. În schimb, serverul X convertește codul cheie în caractere semnificative folosind un proces din două etape:

- În prima etapă, serverul X convertește codul tastă într-un nume simbolic, cunoscut sub numele *keysym*. Toate combinațiile semnificative între o tastă și modificatorii acesteia dispun de *keysym*-uri unice, care sunt constante definite în fișierul */usr/include/X11/keysymdef.h* (acest fișier este destinat utilizării în aplicațiile X). *keysym*-ul rezultat în urma unei apăsări de tastă depinde de starea tastelor modificador, dar și de tasta în sine. Dacă apăsați numai tasta A, veți obține un a cu literă mică. Dacă apăsați tasta A în timp ce tasta Shift este apăsată, rezultatul va fi un A cu majusculă. *keysym*-ul este cel care face diferența între cele două situații și atribuie numele *XK_a*, respectiv *XK_A*, pentru A cu literă mică, respectiv majusculă.

- În a doua etapă, serverul X convertește *keysym*-ul într-un șir text ASCII care poate fi folosit de aplicația X pentru afișare (și salvare în fișiere). Pentru majoritatea tastelor, acest șir are un singur caracter, dar tastele funcționale (mai ales cele programabile) pot genera șiruri de caractere multiple.

Puteți folosi utilitarul *xmodmap* pentru a modifica maparea codurilor tastă în *keysym*-uri. Să presupunem că doriți să inversați între ele semnificațiile tastelor A și Q. Pentru a realiza acest lucru în X, inserați următoarele linii într-un fișier denumit *xmodtest*:

```
keycode 38 = Q
keycode 24 = A
```

Pentru a pune aceste definiții în aplicare, introduceți următoarea comandă într-o fereastră *xterm*:

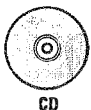
```
xmodmap xmodtest
```

După execuția acestei comenzi, veți descoperi că tastele A și Q au fost inversate.

Fișierul *xmodtest* reprezintă o simplă hartă de tastatură sub X. Ca și în cazul hărții de tastatură Linux, termenul cod tastă reprezintă o tastă fizică. Cu toate acestea, numărul X aferent unei taste fizice este diferit de numărul Linux pentru tasta respectivă. De regulă, codurile tastă X sunt cu opt unități mai mari decât codurile tastă corespondente din Linux.

Întrebări specifice legate de tastatură

Când tasta Num Lock este activă, puteți folosi minitastatura numerică pentru a introduce numere. Deoarece tastatura cu 101 taste deja dispune de taste cursor și de taste Page Up și Page Down, puteți folosi minitastatura numerică numai pentru introducerea numerelor.



CD

Dacă doriți să folosiți minitastatura numerică în acest mod, trebuie să activați tasta Num Lock la lansarea sistemului. Pentru aceasta, trebuie să folosiți programul `setleds` (setare a LED-urilor) inclus în pachetul `kbd` existent pe CD-ROM. Pachetul `kbd` este instalat o dată cu instalarea Linux de pe CD-ROM-ul anexat acestei cărți.

Programul `setleds` modifică setările aferente tastelor Num Lock, Caps Lock și Scroll Lock. Pentru a activa Num Lock la inițializarea sistemului, folosiți `setleds` într-unul din fișierele script rulate de procesul `init` la lansarea Linux. Puteți activa tasta Num Lock (și LED-ul asociat) prin inserția următoarelor comenzi de prompt din fișierul `/etc/rc.d/rc.local`:

```
for t in 1 2 3 4 5 6 7 8
do
    setleds +num < /dev/tty$t > /dev/null
done
```

Opțiunea `+num` determină programul `setleds` să activeze tasta Num Lock; comanda `setleds -num` determină dezactivarea tastei respective.



Secret

Tastatura Gateway AnyKey include o funcție încorporată de remapare a tastelor la un nivel și mai jos decât driverul de tastatură Linux. Dacă această situație ridică probleme, trebuie să apăsați combinația de taste Ctrl-Alt-Suspend_Macro pentru a reseta tastele la maparea normală.

Mouse-ul în Linux

Deși Linux vă permite să folosiți un mouse în modul de afișare text, vă puteți descurca și fără mouse dacă intenționați să vă limitați la afișajele mod text. Dacă doriți să folosiți interfața grafică „indică și clic” furnizată de sistemul X Window System, trebuie să dispuneți de un mouse. În realitate, serverul X nu va fi lansat dacă nu este găsit mouse-ul, deci vom discuta despre mouse în contextul utilizării sale în XFree86.

Interfețe de mouse

Ca și în cazul unităților CD-ROM, interfața de mouse este mai importantă decât producătorul și modelul mouse-ului. Pentru a include suport pentru un mouse de magistrală, este necesar să re-generați nucleul. Când configurați nucleul (vezi capitolul 2 pentru a vedea procedura necesară), programul de configurare vă cere să identificați tipul de interfețe de mouse pe care doriți să le includeți în nucleu. În continuare sunt prezentate întrebările de configurare referitoare la mouse:

```
ATIXL busmouse support (CONFIG_ATIXL_BUSMOUSE) [Y/m/n/?] n
Logitech busmouse support (CONFIG_BUSMOUSE) [Y/m/n/?] n
Microsoft busmouse support (CONFIG_MS_BUSMOUSE) [Y/m/n/?] m
PS/2 mouse (aka "auxiliary device") support (CONFIG_PSMOUSE) [Y/m/n/?] y
C&T 82C710 mouse port support (as on TI Travelmate)
(CONFIG_82C710_MOUSE) [Y/n/?] n
```

În esență, aceste întrebări indică tipurile de interfețe de mouse suportate de Linux.

În afară de utilizarea acestor tipuri speciale de interfețe de mouse, puteți conecta un mouse la PC prin intermediul unui port serial. Astfel, Linux oferă următoarele interfețe de mouse:

- Interfață serială. În esență, mouse-ul este conectat la un port serial, exact ca un modem; dispozitivele tip mouse produse de Microsoft, Logitech și Mouse Systems sunt disponibile cu o interfață serială.
- Port PS/2 Auxiliary Device. Această interfață de mouse funcționează prin controllerul de tastatură.

- Microsoft Busmouse. Acest mouse mai este cunoscut sub numele de *mouse InPort* — se conectează la o placă de interfață care se conectează la magistrala din placa de bază a calculatorului dumneavoastră.
- Logitech Busmouse. Acest mouse este similar mouse-ului de magistrală Microsoft, dar respectă un alt protocol: standardul pentru schimbul de date dintre mouse și placa de bază.
- ATI-XL Busmouse. Acest mouse este o variantă a mouse-ului de magistrală Microsoft, cu excepția faptului că mouse-ul de magistrală ATI-XL vine cu placa video ATI-XL, care conține o interfață încorporată de mouse.
- Chips & Technologies C&T 82C710. Aceasta este interfața QuickPort folosită pe unele laptop-uri, cum sunt Toshiba și mai vechile TravelMate produse de Texas Instruments. Interfața C&T 82C710 se bazează pe interfața PS/2. Este necesară includerea în nucleu a suportului pentru ambele tipuri de interfață.

Majoritatea calculatoarelor folosesc pentru mouse interfața PS/2 Auxiliary Device. Printre dispozitivele mouse de magistrală, interfața Microsoft Busmouse este cea mai populară.

Dacă dispuneți de un mouse de magistrală sau de un mouse PS/2, nucleul Linux identifică mouse-ul la inițializare. Puteți vedea numele mouse-ului în mesajele de boot afișate de nucleu. Examinați aceste mesaje folosind comanda `dmesg | more` și căutați tipul de mouse identificat de nucleu. (Mai puteți folosi comanda `more /var/log/messages` pentru a parcurge mesajele de boot.)

Toate interfețele de mouse necesită utilizarea unui IRQ, iar unele au nevoie de o adresă de port I/O. Porturile seriale COM1 și COM2 folosesc respectiv IRQ-urile 4 și 3. Pentru un mouse de magistrală, nucleul folosește un IRQ prestabilit de 5, care poate intra în conflict cu setările de întrerupere ale altor periferice, cum sunt plăcile de sunet și controllerele SCSI.



Secret

Portul PS/2 Auxiliary folosește întotdeauna IRQ 12. Această setare nu poate fi modificată, deci dacă vreun alt periferic încearcă să folosească IRQ 12, trebuie să modificați IRQ-ul perifericului respectiv.

Interfețe pentru mouse de magistrală tip Microsoft și Logitech

Acest tip de mouse se leagă la o placă de interfață care se conectează la magistrală pe placa de bază a calculatorului dumneavoastră. Cablul de mouse are un conector rotund cu nouă pini și un canal pe o parte. Acest tip de conector este cunoscut sub numele de *conector mini-DIN cu nouă pini*.

Plăcile video ATI Graphics Ultra și Ultra Pro includ un port de mouse compatibil cu interfața pentru mouse de magistrală de tip Logitech. Dacă aveți un mouse conectat la aceste plăci video, specificați-l ca mouse de magistrală de tip Logitech. În ciuda numelui ATI, aceste interfețe de mouse nu sunt interfețe de mouse de magistrală ATI-XL.

Interfața PS/2

Interfața de mouse PS/2 nu necesită nici o placă de extensie. În schimb, mouse-ul se conectează la portul PS/2 Auxiliary Device al controllerului de tastatură. Portul de mouse PS/2 folosește un conector mini-DIN cu șase pini, exact ca și conectorul de tastatură. Perifericul PS/2 Auxiliary este folosit pe scară largă în majoritatea sistemelor de calcul de dată recentă. De asemenea, interfața PS/2 este folosită la unele sisteme laptop.

Mouse-ul ATI-XL

Plăcile video ATI-XL conțin o interfață de mouse cunoscută sub numele de mouse de magistrală ATI-XL. Nu trebuie să vă faceți probleme cu acest tip de mouse decât dacă dispuneți de o placă de interfață ATI-XL.

Numele de periferic ale dispozitivelor de mouse

Dispozitivul de mouse este un dispozitiv de tip caracter similar cu tastatura și cu modemul. *Dispozitivele de tip caracter* transferă informația caracter cu caracter sau mai multe caractere simultan, spre deosebire de *dispozitivele de tip bloc* (cum sunt hard-discurile) care transferă datele sub formă de blocuri de dimensiune fixă. Pentru fiecare interfață de mouse suportată, Linux folosește un nume de periferic special, după cum urmează:

- mouse de magistrală ATI-XL: /dev/atibm
- mouse de magistrală Microsoft InPort: /dev/inportbm
- mouse de magistrală Logitech: /dev/logibm
- port PS/2 Auxiliary pentru mouse: /dev/psaux
- nume de periferice de port serial pentru COM1 și COM2 (folosite pentru mouse serial): /dev/ttyS0 și /dev/ttyS1

Prin convenție, perifericul de mouse generic /dev/mouse este setat sub forma unei legături la perifericul de mouse efectiv instalat în sistemul dumneavoastră. Pe sistemul meu, care dispune de un port de mouse Auxiliary PS/2, un listing detaliat al fișierului /dev/mouse se prezintă astfel:

```
ls -l /dev/mouse
lrwxrwxrwx 1 root root 5 Jan 25 20:41 /dev/mouse -> psaux
```

Acest listing arată că /dev/mouse este o legătură simbolică spre /dev/psaux, care reprezintă interfața PS/2 Auxiliary Device.

Pe de altă parte, pe vechiul meu sistem 386 PC, mouse-ul se află instalat la primul port serial. În acest caz, listingul detaliat al fișierului /dev/mouse arată astfel:

```
ls -l /dev/mouse
lrwxrwxrwx 1 root root 5 Jan 25 20:41 /dev/mouse -> ttyS0
```

Convenția de conectare a perifericului /dev/mouse la perifericul de mouse efectiv instalat în sistem permite acelor programe care folosesc /dev/mouse să facă referire la perifericul de mouse, fără a le interesa tipul real de mouse. Datorită legăturii simbolice între /dev/mouse și perifericul real, toate cererile de intrare și de ieșire ajung direct la driverul de mouse, care știe cum să trateze cererile respective.

Protocoloale de mouse

Cu atâtea tipuri de interfețe și nume de periferic, probabil că sunteți destul de derutat cu privire la utilizarea unui mouse în Linux. Mai aveți însă nevoie de o altă informație pentru a descrie complet mouse-ul pentru XFree86, și anume protocolul de mouse.

Protocolul de mouse este convenția folosită de mouse pentru împachetarea informației despre deplasarea mouse-ului și stările butoanelor (apăsate sau libere). Puteți asimila interfața unui mouse (serială, de magistrală sau PS/2 Auxiliary) ca fiind mecanismul fizic de schimb de date. Protocolul de mouse, pe de altă parte, este folosit de receptorul datelor primite de la mouse (starea butonului sau deplasarea) pentru a descifra datele respective și pentru a întreprinde o acțiune (cum ar fi deplasarea cursorului pe ecran).

Protocolul de mouse este important pentru XFree86, deoarece serverul X trebuie să interpreteze și să folosească datele de mouse. Veți afla mai multe despre protocoale în secțiunea următoare.

Mouse-ul și XFree86

În XFree86, receptorul datelor de mouse este serverul X. Informația de mouse minimală necesară serverului X este compusă din două elemente:

- Numele de periferic al mouse-ului. (Puteți specifica /dev/mouse, cu condiția ca acesta să fie legat în mod simbolic cu perifericul de mouse efectiv instalat.)
- Protocolul folosit de mouse pentru a trimite rapoarte asupra mișcării mouse-ului și a poziției butoanelor.

Pentru anumite tipuri de mouse, cum este mouse-ul serial Logitech, serverul X mai necesită informații precum BaudRate (viteză de baud) sau SampleRate (frecvență de eșantionare). De asemenea, mai puteți specifica și alte informații, precum modul de simulare al celui de-al treilea buton pe un mouse cu două butoane (folosit la majoritatea calculatoarelor personale).

Serverul X se așteaptă să primească această informație din fișierul /etc/XF86Config, adică fișierul de configurare pentru XFree86. Concret, secțiunea Pointer din acest fișier specifică perifericul și protocolul de mouse. Dacă dispuneți de un mouse serial Microsoft conectat la portul COM1 (primul port serial al unui PC), o secțiune Pointer elementară din fișierul /etc/XF86Config se prezintă astfel:

```
Section "Pointer"
    Protocol "Microsoft"
    Device "/dev/ttyS0"
EndSection
```

Linia Device specifică numele de periferic al mouse-ului.

Specificații de protocol

Linia Protocol indică protocolul de mouse. În continuare este prezentată sintaxa liniei Protocol:

Protocol "nume-protocol"

nume-protocol este numele unui protocol de mouse. (Ghilimelele duble sunt necesare.)

În tabelul 13.2 sunt prezentate numele de protocol de mouse recunoscute de XFree86.

Tabelul 13.2 Nume de protocol de mouse recunoscute de XFree86

Nume protocol	Tip de mouse corespunzător
BusMouse	Protocol folosit pentru mouse-ul de magistrală de tip Microsoft și Logitech.
Logitech	Protocol folosit pentru dispozitivele mouse serial de tip Logitech mai vechi. Dispozitivele de tip mouse mai recente produse de Logitech folosesc protocolul Microsoft sau Mouseman.
Microsoft	Protocol folosit pentru mouse-ul Microsoft și alte dispozitive de tip mouse serial.
MMSeries	Protocol folosit pentru dispozitivele mouse serial produse de MMSeries.
Mouseman	Protocol folosit de mouse-ul Logitech Mouseman.
MouseSystems	Protocol folosit de dispozitivele mouse produse de Mouse Systems.

(continuare)

Tabelul 13.2 Continuare

Nume protocol	Tip de mouse corespunzător
PS/2	Protocol folosit pentru orice mouse conectat la portul PS/2 Auxiliary.
MMHitTab	Protocol folosit pentru MouseMan HitTablet.
Xqueue	Protocol folosit numai dacă protocolul keyboard este de asemenea setat la Xqueue.
OSMouse	Protocol destinat pentru SCO UNIX (deoarece XFree86 este funcțional și sub SCO UNIX, versiune comercială a sistemului UNIX).

Alte informații de configurare a mouse-ului

O altă opțiune care poate fi specificată în secțiunea Pointer este modalitatea de manipulare a celui de-al treilea buton de mouse. Multe tipuri populare de periferice tip mouse pentru PC-uri dispun de numai două butoane, dar aplicațiile X presupun frecvent existența a trei butoane. O opțiune este efectuarea unui clic simultan pe ambele butoane ale mouse-ului pentru a simula butonul din mijloc. Pentru a simula un clic cu butonul din mijloc printr-un clic simultan pe butoanele din stânga și din dreapta, trebuie să specificați următoarea linie în secțiunea Pointer a fișierului XF86Config:

```
Emulate3Buttons
```

Unele periferice de tip mouse cu trei butoane – cum este Logitech Mouseman – trimit un clic simultan pe butoanele din stânga și din dreapta la efectuarea unui clic pe butonul din mijloc. Pentru a informa serverul X cu privire la această situație, trebuie să inserați următoarea linie în fișierul de configurare:

```
ChordMiddle
```

Cu alte cuvinte, chiar dacă dispuneți de un mouse cu trei butoane, este posibil ca acesta să nu semnaleze stările a trei butoane distincte. Este necesar să folosiți indicatorul ChordMiddle pentru a vă asigura că serverul X tratează în mod corect efectuarea unui clic cu butonul din mijloc.



Sugestie

Pentru perifericele de tip mouse serial produse de Logitech, mai trebuie să specificați viteza de baud (viteza cu care mouse-ul trimite datele) și frecvența de eșantionare (numărul de evenimente de mouse care survin într-o secundă). În

continuare este prezentat un set obișnuit de parametri:

```
BaudRate 9600
SampleRate 150
```

Alte două intrări din secțiunea Pointer a fișierului XF86Config sunt destinate perifericelor de tip mouse care pot opera în două moduri de protocol. Ambele linii prezentate mai jos trebuie specificate pentru un mouse MouseSystems:

```
ClearDTR
ClearRTS
```

Utilizarea unui mouse în X

Un mouse iese spectaculos în evidență în aplicațiile X, deoarece X este un sistem bazat pe ferestre grafice. Modul de utilizare a mouse-ului depinde de fiecare aplicație în parte. În Linux, de regulă nu există multe aplicații X (prin aplicații se înțeleg programe precum procesoare de text sau programe de calcul tabelar). De regulă, utilizatorul dispune de un program de gestionare a ferestrelor, precum `fvwm2`, și una sau mai multe ferestre `xterm` pe ecran. Dată fiind „moda” actuală a browserelor Web, este posibil ca și unul dintre aceste programe să fie în funcțiune.

Majoritatea aplicațiilor grafice răspund la evenimentele de mouse cum sunt clicurile și dublu clicurile. Reacția exactă depinde de aplicația care a recepționat evenimentul de mouse. Cu un mouse se pot efectua numeroase operații. În continuare sunt prezentate cele șase operații elementare folosite pentru interacțiunea cu o aplicație grafică X:

- **Apăsați** un buton de mouse prin apăsarea butonului în jos fără a deplasa mouse-ul.
- **Eliberați** un buton de mouse pe care l-ați ținut apăsat. Prin eliberarea butonului se inițiază, de regulă, o acțiune oarecare.
- **Efectuați clic** pe un buton de mouse prin apăsarea și eliberarea rapidă a acestuia.
- **Efectuați dublu clic** pe un buton de mouse prin efectuarea rapidă a două clicuri într-o succesiune rapidă, fără a deplasa mouse-ul.
- **Trageți** indicatorul de mouse prin apăsarea unui buton și deplasarea mouse-ului cu butonul apăsat.
- **Deplasați** indicatorul de mouse prin deplasarea mouse-ului fără a apăsa nici un buton.

Butonul din stânga al mouse-ului este folosit pentru a indica o selecție. Dacă o interfață grafică afișează un buton de presiune, acel buton se activează prin efectuarea unui clic cu butonul din stânga al mouse-ului.

Programul de gestionare a ferestrelor `fvwm2` afișează un meniu pop-up – câte unul pentru fiecare buton – la efectuarea unui clic pe un buton de mouse, în timp ce indicatorul de mouse se află în fereastra root a sistemului X. Cadrul din jurul fiecărei ferestre aparține, de asemenea, programului de gestionare a ferestrelor. Astfel, acest program controlează ceea ce se întâmplă la efectuarea unui clic pe cadrul ferestrei prin efectuarea unui clic pe orice buton de mouse. De exemplu, dacă efectuați un clic în colțul din dreapta-sus al unui cadru de fereastră, `fvwm2` afișează un meniu care vă permite să deplasați fereastra, să o redimensionați, să o reduceți la o pictogramă sau chiar să o distrugeți.

Dacă ați folosit Microsoft Windows sau un calculator Apple Macintosh, probabil că operația de tăiere și lipire (cut-and-paste) vă este familiară. Puteți selecta text într-o aplicație, îl tăiați, vă deplasați în fereastra unei alte aplicații și îl lipiți acolo.

Și X suportă conceptul de tăiere și lipire, dar etapele exacte depind de setul de instrumente (toolkit) X folosit pentru generarea unei aplicații. Tăierea și lipirea din Motif (o interfață grafică populară) poate fi diferită sub OPEN LOOK.



Sugestie

Aplicația `xterm` include suport pentru tăiere și lipire într-o manieră mai simplă. Dacă mențineți apăsat butonul din stânga al mouse-ului și trageți cu mouse-ul peste orice text din fereastra `xterm`, textul respectiv devine selecția curentă. Apoi, dacă efectuați clic pe butonul din mijloc al mouse-ului în aceeași fereastră `xterm` sau într-o alta, selecția respectivă este lipită.

Eroarea device not found

La lansarea sistemului X (folosind comanda `startx`), serverul X poate eșua, afișând ulterior un mesaj de genul:

```
device not found (/dev/mouse or /dev/psaux)
```

Acest mesaj înseamnă, de regulă, faptul că fișierul de periferic (de exemplu `/dev/psaux`) nu există pentru perifericul de tip mouse atașat la sistem.

Eroarea `device not found` (perifericul nu a fost găsit) poate surveni dacă specificați în mod eronat tipul de mouse la instalarea sistemului Linux. Să presupunem că folosiți un mouse

Microsoft, care se conectează în portul PS/2 Auxiliary. În acest caz, trebuie să specificați tipul de mouse ca fiind PS/2, *nu* Microsoft (chiar dacă Microsoft este producătorul mouse-ului). Nu uitați că ceea ce contează este tipul interfeței de mouse, nu producătorul și modelul mouse-ului.

Eroarea device busy

Dacă primiți o eroare `device busy` (periferic ocupat) când încercați să lansați X, înseamnă că un alt program folosește deja perifericul de mouse. Un potențial vinovat mai poate fi programul `gpm`, care vă permite să folosiți mouse-ul în mod text.



Secret

Dacă dispuneți de un mouse de magistrală, asigurați-vă că nu rulați programul `gpm` (acest program vă permite să decupați și să inserați text în modul de afișare text).

Pentru a verifica dacă `gpm` rulează, folosiți următoarea comandă și căutați un proces `gpm` în datele de ieșire:

```
ps ax | grep gpm
```

Dacă `gpm` rulează, folosiți comanda `gpm -k` pentru a opri programul înainte de a inițializa sistemul X.

Înlocuitoare de mouse

Deși nu am discutat în mod explicit despre alte periferice de indicare, Linux și XFree86 pot accesa și folosi și alte periferice de indicare, cu condiția ca acestea să se comporte ca un mouse.

De exemplu, noul periferic de indicare ALPS GlidePoint funcționează sub Linux și X. Tot ce aveți de făcut este să îl setați ca pe un mouse serial Microsoft.

Când întâlniți alte periferice de indicare, singura dumneavoastră problemă va fi de a afla dacă dispozitivul respectiv este compatibil cu vreunul dintre tipurile de mouse utilizate de Linux și X.

Multe sisteme laptop folosesc dispozitive de tip trackball în loc de un mouse. Acestea sunt frecvent compatibile cu mouse-ul PS/2. Astfel, setați un asemenea trackball ca pe un mouse PS/2 și apoi încercați să-l utilizați.

Rezumat

Linux acceptă orice tastatură care funcționează sub MS-DOS, iar dumneavoastră aveți posibilitatea de a configura tastatura și pentru alte limbi în afară de limba engleză. Mouse-ul nu este frecvent folosit în mod text, dar XFree86 X Window System necesită un mouse pentru a putea rula. XFree86 include suport pentru principalele tipuri de mouse existente. Acest capitol descrie modul de utilizare și configurare a tastaturii și a mouse-ului în Linux și X.

După lectura acestui capitol, ați învățat următoarele:

- ▶ Tastatura are doi parametri configurabili: intervalul de repetare (adică intervalul de așteptare anterior începutului repetiției afișării unui caracter pe ecran) și viteza de repetare (viteza de repetiție a unui caracter când tasta corespunzătoare se menține apăsată). Acești parametri pot fi setați cu programul `kbdrate` în Linux.
- ▶ Pentru a putea folosi calculatorul și în alte limbi decât limba engleză, trebuie să modificați maparea tastaturii, în speță modul în care o tastă fizică este interpretată ca un caracter dintr-un alfabet. De asemenea, trebuie să determinați diferențele aplicații să accepte și să afișeze caracterul. Multe limbi folosesc accente pe care maparea standard

a tastaturii le ignoră. Cu toate acestea, puteți folosi programul `loadkeys` în mod text și `xmodmap` în X pentru a modifica maparea tastaturii.

- ▶ Mouse-ul este o necesitate când doriți să rulați XFree86, sistemul X Window System pentru Linux. XFree86 include suport pentru majoritatea tipurilor populare de interfețe de mouse, cum sunt porturile seriale, mouse-ul de magistrală și portul PS/2 Auxiliary. Interfața de mouse se identifică printr-un dispozitiv precum `/dev/psaux` pentru portul PS/2, respectiv `/dev/ttyso` pentru portul serial COM1.
- ▶ *Protocolul de mouse* se referă la modul de formatare a datelor de mouse (modul în care mouse-ul semnalează mișcările și stările butoanelor). XFree86 include suport pentru multiple protocoale de mouse frecvent folosite: mouse de magistrală, PS/2, Microsoft și Logitech.
- ▶ În fișierul de configurare XFree86, `/etc/XF86Config`, trebuie să furnizați informații despre tastatură și mouse în secțiunile `Keyboard` și `Pointer`.
- ▶ În afară de perifericele de tip mouse, dispozitivele de indicare (cum sunt cele de tip trackball și noul GlidePoint) trebuie să funcționeze în X, dacă interfața și protocolul sunt compatibile cu cele ale unui tip de mouse suportat de sistemul XFree86.

Capitolul 14

Imprimante

În acest capitol

- ▶ Veți învăța ce înseamnă tipărirea din punctul de vedere al utilizatorului
- ▶ Veți învăța despre derulare (spooling) – conceptul cozii de tipărire
- ▶ Veți configura mediul de tipărire în Linux
- ▶ Veți tipări fișiere PostScript folosind Ghostscript
- ▶ Veți configura o nouă imprimantă în Linux

Când configurați Linux pe un calculator personal, probabil că imprimanta este ultimul lucru la care vă gândiți. Mai întâi de toate, doriți ca Linux să ruleze pe calculatorul dumneavoastră. Apoi vă decideți să puneți modemul în funcțiune pentru a vă apela calculatorul de la birou sau furnizorul dumneavoastră de servicii Internet. Pe măsură ce dependența dumneavoastră de Linux crește, veți dori să tipăriți texte sau fișiere PostScript primite din rețea, moment în care veți dori să știți cum să vă folosiți imprimanta sub Linux.

După cum probabil ați intuit, conectarea fizică a unei imprimante la portul paralel al unui calculator personal este relativ simplă și nu depinde de sistemul de operare. Configurarea programului de tipărire este cea care necesită un oarecare efort. Ca atare, capitolul de față furnizează informații referitoare la setarea mediului de tipărire pentru Linux.

Calculatorul personal, imprimanta și Linux

În mod obișnuit, calculatoarele personale dispun de un port paralel și două porturi seriale. *Portul paralel* este denumit astfel deoarece poate transfera 8 biți de date în paralel. *Portul serial* trebuie să trimită date într-un mod serial, orientat pe biți. De exemplu, un bait este transferat sub forma unei secvențe de 8 biți care parcurg portul serial unul după altul. Avantajul este că portul paralel este cu mult mai rapid decât portul serial.

Deși o imprimantă poate fi conectată la un PC printr-un port serial sau paralel, majoritatea utilizatorilor preferă să conecteze imprimanta la portul paralel al calculatorului personal.

Numele de periferic ale imprimantelor

În MS-DOS, primul port paralel se numește *LPT1*, iar al doilea se numește *LPT2*. Porturile seriale se numesc *COM1* și *COM2*.

În Linux, porturile paralele au nume de periferic similare altor periferice, cum sunt */dev/ttyS0* pentru primul port serial al calculatorului sau */dev/hda* pentru prima unitate de disc IDE.

Numele */dev/lp0* și */dev/lp1* desemnează primul, respectiv al doilea port paralel.

Atenție, totuși: chiar dacă sistemul dumneavoastră dispune de un singur port paralel, aceasta nu înseamnă că Linux va folosi */dev/lp0* ca nume de periferic pentru portul paralel respectiv. Când Linux este inițializat, acesta afișează informații despre portul paralel detectat. Examinați mesajele de boot cu comanda `dmesg | more` pentru a vedea care este portul paralel detectat de

Linux. Pe calculatorul meu, care dispune de un singur port paralel, mesajul de boot conține următoarea linie:

```
lp1 at 0x0378, using polling driver
```

Astfel, pentru calculatorul meu Linux PC, portul paralel este */dev/lp1*.

Tabelul 14.1 prezintă lista porturilor paralele și numerele de periferic din Linux:

Tabelul 14-1 Numele de periferic și numerele porturilor paralele

Adresă I/O	Nume de periferic	Număr major	Număr minor
0x3bc	<i>/dev/lp0</i>	6	0
0x378	<i>/dev/lp1</i>	6	1
0x278	<i>/dev/lp2</i>	6	2

Pe majoritatea calculatoarelor personale, portul LPT1 folosește adresa 0x378. Astfel, numele de periferic al portului paralel este */dev/lp1*, care este numele de periferic folosit pentru o imprimantă conectată la portul paralel.

Derularea și lucrările de tipărire

Probabil că deja sunteți familiarizat cu conceptul de derulare (spooling) de la tipărirea sub Microsoft Windows. *Derularea* se referă la posibilitatea de tipărire în fundal. Când tipăriți dintr-un procesor de texte în Windows, de exemplu, rezultatul tipăririi ajunge mai întâi la un fișier pe disc. Apoi, când continuați lucrul cu procesorul de texte, un proces de fundal trimite ieșirea respectivă (fișierul) la imprimantă.

Mediul de tipărire Linux, alcătuit din numeroase programe pe care le vom descrie ulterior în acest capitol, include de asemenea suport pentru derulare. Termenul *director de derulare* (*spool directory*) se referă la directorul care conține fișierele de ieșire destinate imprimantei.

Lucrarea de tipărire (*print job*) se referă la ceea ce se tipărește cu o singură comandă de tipărire. Mediul de tipărire așază la coadă lucrările de tipărire prin stocarea acestora în directorul de derulare. Apoi, un proces de fundal poate trimite periodic lucrările de tipărire din directorul de derulare la imprimantă.



Mediul de tipărire Linux a evoluat din facilitățile de tipărire ale BSD (Berkeley Software Distribution) UNIX. Pe măsură ce dobândiți mai multe cunoștințe despre UNIX (sau dacă sunteți un veteran UNIX) veți descoperi utilitatea acestei informații, deoarece vă indică comenzile de tipărire disponibile în Linux. Dacă nu știți nimic despre BSD UNIX, nu vă faceți griji. Vom detalia comenzile de tipărire în secțiunea următoare.

Tipărirea în Linux din punctul de vedere al utilizatorului

Înainte de a descrie modul de configurare a sistemului Linux în vederea tipăririi, vom trece în revistă comenzile de tipărire elementare. Nu veți putea testa comenzile de tipărire înainte de a configura mediul de tipărire, dar secțiunea de față oferă o imagine a modului de tipărire în Linux (sau în orice sistem BSD UNIX).

Tipărirea cu lpr

În Linux, comanda `lpr` se folosește pentru a așeza o lucrare de tipărire la coadă. Pentru a tipări fișierul `rfc1789.txt`, veți tasta următoarele:

```
lpr rfc1789.txt
```

Comanda `lpr` copiază fișierul `rfc1789.txt` într-un director de derulare (localizat în directorul `/var/spool/lpd/lp`). Periodic, un program de tipărire cunoscut sub numele `lpd` trimite fișierul din directorul de derulare la imprimantă.

Comanda `lpr` poate fi îmbunătățită cu o serie de opțiuni. O opțiune comună este indicarea tipului de imprimantă folosind comutatorul `-P`. Dacă dispuneți de o imprimantă LaserJet Hewlett-Packard denumită `hplj` (în secțiunea „Fișierul `/etc/printcap`”, vom vedea că numele imprimantei apare în fișierul `/etc/printcap`), atunci veți tipări cu următoarea comandă:

```
lpr -Phplj rfc1789.txt
```



Sugestie

În afară de opțiunea `-P` a comenzii `lpr`, puteți specifica imprimanta prestabilită prin intermediul variabilei de mediu `PRINTER` (imprimantă). Dacă numele imprimantei dumneavoastră este `hplj`, veți folosi următoarea comandă pentru a vă sigura că `lpr` trimite toate lucrările de tipărire la imprimanta `hplj`:

```
PRINTER=hplj; export PRINTER
```



Puteți insera definiția de mediu în script-ul de login prestabilit pentru Bash. Pentru aceasta, adăugați următoarea linie în fișierul `/etc/profile`:

```
export PRINTER=hplj
```

În acest exemplu s-a presupus că numele imprimantei conectate la sistemul dumneavoastră este `hplj`. Numele respectiv trebuie înlocuit cu numele pe care l-ați atribuit imprimantei din sistemul dumneavoastră (Veți învăța să denumiți o imprimantă în secțiunea intitulată „Numele imprimantei“.)

Deoarece scriptul de login prestabilit — `/etc/profile` — este comun tuturor login-urilor care folosesc interpretorul bash, definirea variabilei de mediu `PRINTER` în `/etc/profile` scutește utilizatorii de efortul de a specifica o imprimantă în mod explicit.

Examinarea cozii de tipărire cu comanda lpq

Când `lpr` așază o lucrare de tipărire la coadă, nu afișează mesaje. Dacă tipăriți din greșală un fișier mare și doriți să opriți lucrarea de tipărire înainte de a risipi prea multă hârtie, trebuie să folosiți comanda `lpq` pentru a examina lucrările de tipărire curente. În continuare este prezentat un listing specific al lucrărilor de tipărire, obținut cu comanda `lpq`:

```
lpq
lp is ready and printing
Rank      Owner      Job      Files      Total Size
active    root       7        sendmail.cf 30192 bytes
```

Cuvântul `active` din coloana `Rank` indică lucrarea aflată la tipărire în momentul respectiv. Restul intrărilor indică lucrări în ordinea în care vor fi tipărite. Dacă nu vă vedeți lucrarea în listă, înseamnă că tipărirea acesteia s-a încheiat.

Anularea lucrării de tipărire cu lprm

Pentru a elimina o lucrare din coada de tipărire, folosiți comanda `lprm`. Pentru a opri lucrarea de tipărire nr.1, tastați următoarele:

```
lprm 1
dFA001Aa00235 dequeued
cFA001Aa00235 dequeued
```



Sugestie

Dacă vă grăbiți și doriți să anulați toate lucrările de tipărire pe care le-ați trimis până acum, folosiți `lprm` cu un semn minus ca argument, după cum urmează:

```
lprm -
```

Examinarea stării imprimantei cu comanda lpc status

Pentru a vedea numele imprimantelor conectate la sistemul dumneavoastră, folosiți comanda `/usr/sbin/lpc status`. Pe un sistem obișnuit, comanda `lpc status` poate afișa următoarele:

```
/usr/sbin/lpc status
lp:
    queuing is enabled
    printing is enabled
    no entries
    no daemon present
```

Acest exemplu de ieșire prezintă starea unei imprimante denumite `lp`, adică numele de imprimantă prestabilit folosit de comenzi de tipărire precum `lpr`, `lpq` și `lprm` când nu se specifică în mod explicit un nume de imprimantă.



Cuvântul `daemon` din ultima linie a mesajului de stare se referă la procesul de fundal (denumit `lpd`) care se ocupă de tipărirea efectivă. În UNIX, termenul `daemon` este folosit pentru procese de fundal care monitorizează și execută multe funcții de sistem de o importanță esențială. De regulă, un `daemon` este lansat la încărcarea sistemului de operare, iar `daemonul` procesează rularea atâta timp cât sistemul funcționează. Majoritatea `daemonilor` au posibilitatea de a restarta copii ale lor înșiși pentru a executa anumite taskuri. De asemenea, deși nu este o regulă, majoritatea `daemonilor` au nume care se termină cu `d`, precum `crond`, `syslogd`, `klogd`, `inetd`, `lpd` (acesta este `daemonul` de tipărire), `named` și `httpd`.

Nu trebuie să vă îngrijorați dacă ieșirea comenzii `lpc status` nu semnalează prezența nici unui `daemon`. `Daemonul` care controlează coada de tipărire a unei anumite imprimante dispare când directorul de derulare este vid, dar întotdeauna există o copie master a `daemonului` de tipărire, `lpd`, care monitorizează directoarele de tipărire. Când tipăriți ceva la o anumită imprimantă folosind comanda `lpr`, copia master a `daemonului` `lpd` creează un nou proces `lpd` care se ocupă de coada de tipărire a imprimantei respective.

Tipărire cu efecte speciale

Până acum v-am prezentat comenzile de trimiteră a unui fișier la imprimantă. Dacă folosiți o imprimantă PostScript, îi puteți trimite un fișier PostScript pentru a genera un rezultat cu aspect îngrijit.

Dacă doriți să tipăriți un fișier text cu elemente de formatare suplimentare, cum ar fi un antet și un număr de pagină pe fiecare pagină, puteți folosi comanda `pr`. Ideea este de a folosi `pr` pentru

a formata fișierul text și apoi de a trimite rezultatul către lpr. În continuare este prezentat un exemplu:

```
pr -h"Web Server Access Statistics" -l60 webstat.txt | lpr
```

Această linie de comandă tipărește fișierul webstat.txt cu textul Web Server Access Statistics adăugat la începutul fiecărei pagini. Opțiunea -l60 setează lungimea fiecărei pagini la 60 de linii.



Referință
încrucișată

Pentru a realiza alte tipuri de formatare, puteți folosi programe de procesare a textelor precum TeX (inclus în Linux pe CD-ROM-ul anexat acestei cărți).

Capitolul 24 descrie aceste sisteme de formatare și explică modul de tipărire a paginilor cu elemente de formatare.

PostScript este în prezent un format comun pentru fișiere de documentație software, mai ales pentru fișierele transferate din Internet. Dacă dispuneți de fișiere PostScript, le puteți previzualiza și tipări folosind programul Ghostview. (Pentru a testa programul, tasteți **ghostview** într-o fereastră xterm. Pentru a rula Ghostview aveți nevoie de sistemul X Window System.)

Tipărirea din fundal

În Linux, controlul exercitat de utilizator asupra procesului de tipărire se bazează pe comenzile elementare lpr, lpq, lprm și lpc status. Pentru a înțelege modul de tipărire din fundal mai aveți nevoie de câteva informații. Ca și multe altele în Linux, suportul pentru tipărire se reduce la existența fișierelor potrivite la locurile potrivite. Acest fapt devine evident după ce ați configurat Linux pentru lucrul în rețea și ați setat modemurile pentru conectarea la rețea prin apel telefonic, de exemplu. În fiecare caz, trebuie să vă asigurați că mai multe fișiere de configurare apar acolo unde trebuie. Mediul de tipărire are același deziderat: fișiere de configurare care specifică modul de desfășurare a tipăririi.

Dacă aveți experiență în utilizarea MS-DOS, ați tipărit fișiere prin simpla copiere a acestora la portul de tipărire (LPT1, de exemplu). Puteți crede că o abordare similară se aplică și pentru Linux. După cum vom vedea în secțiunea următoare, o asemenea metodă „brutală” de copiere a fișierului la portul fizic al imprimantei nu este adecvată pentru un sistem multiutilizator ca Linux. Aveți nevoie de o modalitate de a înșirui la coadă lucrările de tipărire și de un proces de tipărire separat, care să se ocupe de tipărirea în sine.

Copierea la imprimantă: tipărirea prin metoda „brutală”

Dacă aveți o imprimantă conectată la portul paralel /dev/lp1 (puteți găsi această informație din mesajele de boot), puteți tipări un fișier text prin simpla trimitere a fișierului la imprimantă, folosind următoarea comandă:

```
cat webstat.txt > /dev/lp1
```

Această comandă va genera, într-adevăr, o versiune tipărită a fișierului webstat.txt, cu condiția îndeplinirii următoarelor două condiții:

- Trebuie să intrați în sistem ca root (Linux permite numai utilizatorului root și anumitor procese accesul direct la perifericele fizice, cum sunt imprimantele).
- Imprimanta trebuie să fie conectată la portul /dev/lp1, alimentată de la rețeaua de energie electrică și online.

Problema care apare la copierea unui fișier direct la imprimantă este aceea că încheierea comenzii se va realiza numai când copierea (care este echivalentă cu tipărirea, în acest caz) s-a terminat. Pentru un fișier de mari dimensiuni, tipărirea poate dura. În realitate, dacă imprimanta nu este pornită, comanda va părea suspendată (dacă se întâmplă acest lucru, apăsați Ctrl-C pentru a anula comanda).

Derularea – o modalitate mai bună de tipărire

Pe un sistem multitasking și multiutilizator cum este Linux, o modalitate mai indicată de tipărire o constituie *derularea* (spooling) datelor: se trimite o ieșire către un fișier și apoi un proces separat trimite ieșirea la imprimantă. Astfel, vă puteți continua activitatea în timp ce tipărirea are loc în fundal. De asemenea, dacă sistemul dumneavoastră are mai mulți utilizatori, toți aceștia pot tipări la aceeași imprimantă fără a mai fi interesați dacă imprimanta este disponibilă sau nu. Procesul de tipărire în fundal se va ocupa de toate detaliile.

Iată care este modul de funcționare a mediului de tipărire Linux: un director, sau *regiune de derulare*, este configurat pentru fiecare imprimantă. Toate datele destinate unei imprimante sunt stocate în regiunea de derulare a imprimantei respective, câte un fișier pentru fiecare lucrare de tipărire. Un proces de fundal denumit lpd – daemonul de tipărire – verifică în mod constant regiunea de derulare în căutare de fișiere noi. Când lpd găsește un fișier nou, îl trimite la imprimantă.

Regiunea de derulare servește drept coadă pentru lucrările de tipărire, fiecare din acestea reprezentând un fișier din directorul de derulare.



Observație

Deși în mod obișnuit la un calculator personal este conectată la o singură imprimantă, una dintre caracteristicile „de forță” ale sistemului de tipărire Linux și a tipăririi sub BSD UNIX este posibilitatea de a tipări la o imprimantă conectată la un alt sistem din rețea. Tipărirea la o imprimantă de la distanță este manipulată în același mod ca și tipărirea la imprimanta locală, în sensul că există un director de derulare pentru imprimanta îndepărtată. Programul lpd examinează directorul de derulare al imprimantei de la distanță și trimite toate fișierele aflate în așteptare la imprimanta respectivă.

Așa cum s-a explicat anterior, comanda de nivel utilizator pentru derularea unui fișier este lpr. Când lpr rulează, copiază datele din imprimantă într-un fișier din directorul de derulare. Alte comenzi de nivel utilizator pentru lucrul cu cozile de tipărire sunt lpq, lprm și lpc. Așa cum s-a arătat în această secțiune, programul care completează mediul de tipărire Linux este daemonul de tipărire lpd.

În afară de cele cinci programe lpd, lpr, lpq, lprm și lpc, un fișier denumit /etc/printcap are un rol de o importanță crucială în mediul de tipărire Linux. Secțiunea „Fișierul /etc/printcap” descrie fișierul respectiv.

Derularea folosind o legătură simbolică

Când folosiți comanda lpr pentru a derula un fișier în vederea tipăririi, lpr copiază fișierul respectiv în directorul de derulare. Această situație poate reprezenta o problemă la tipărirea unui fișier de mari dimensiuni. Operația de copiere poate consuma timp și veți avea nevoie de spațiu pe disc pentru directorul de derulare (în /usr/spool). Dacă sunteți în criză de spațiu pe disc – mai ales în partiția care folosește arborele directorului /usr – este necesar să evitați operația de copiere.

Comanda `lpr` pune la dispoziția utilizatorului opțiunea `-s`, care determină o *legătură simbolică* în directorul de derulare, determinând ca legătura să facă referire la fișierul pe care doriți să îl tipăriți. Pentru a tipări un fișier de anvergură, puteți folosi comanda `lpr` astfel:

`lpr -s files.index`

În loc de a copia fișierul `files.index` în directorul de derulare, `lpr` creează o legătură simbolică la fișierul respectiv (legătura simbolică va fi amplasată în directorul de derulare).

Puteți crea o legătură simbolică rapid, iar legătura nu folosește aproape deloc spațiul de pe disc, în comparație cu cel potențial folosit de fișier.



Atenție!

Când folosiți comanda `lpr -s` pentru a derula un fișier folosind o legătură simbolică, trebuie să fiți atent la un important efect secundar: deoarece legătura simbolică (din directorul de derulare) este numai o referință la fișierul respectiv, nu puteți edita sau șterge fișierul original decât după încheierea tipăririi.

De asemenea, opțiunea `-s` oprește `lpr` de la copierea fișierului de date în directorul de derulare al sistemului local. Dacă trimiteți o lucrare de tipărire pentru o imprimantă de la distanță, fișierul de date este copiat în directorul de derulare al sistemului de la distanță, indiferent de opțiunea `-s`.

Controlul imprimantei cu comanda `lpc`

Toți utilizatorii pot folosi comanda `lpc status` pentru a examina starea imprimantei. Dacă intrați în sistem ca `root`, puteți folosi `lpc` pentru a executa mai multe funcții de control al imprimantei, cum ar fi pornirea și oprirea derulării, activarea și dezactivarea imprimantelor și modificarea ordinii lucrărilor de tipărire. Puteți folosi al doilea argument al comenzii `lpc` pentru a executa o anumită operație sau pentru a rula `lpc` în mod interactiv.

Pentru a rula `lpc` în mod interactiv, tastați `/usr/sbin/lpc` la prompt-ul interpretorului. La promptul `lpc>`, tastați `help` pentru a vedea o listă de comenzi acceptate de `lpc`. Pentru a obține mai mult ajutor pentru o comandă, tastați `help nume-comanda` (*nume-comanda* este una dintre comenzile `lpc`). În continuare este prezentat un exemplu de sesiune de lucru cu `lpc`:

```
/usr/sbin/lpc
lpc> help
Command may be abbreviated. Commands are:
```

```
abort    enable  disable  help    restart  status  topq    ?
clean    exit   down     quit    start    stop    up
lpc> help status
status    show status of daemon and queue
lpc> q
```



Sugestie

Pentru o anumită operație, puteți folosi pur și simplu sintaxa cu o singură linie, cu `lpc` urmat de una dintre comenzile `lpc` și toate argumentele necesare. Pentru a opri daemonul de derulare a tipăririi (`lpd`) mutați lucrarea de tipărire 39 în partea superioară a cozii, iar pentru a relua derularea veți folosi următoarele comenzi (în exemplu se presupune că numele imprimantei este `hplj`):

```
lpc stop hplj
lpc topq 39
lpc start hplj
```

Tabelul 14-2 prezintă comenzile `lpc` accesibile tuturor utilizatorilor (cu condiția ca directorul `/usr/sbin` să se găsească în variabila de mediu `PATH`; dacă nu, tastați `export PATH=$PATH:/usr/sbin` pentru a actualiza `PATH`).

Tabelul 14-2 Comenzi `lpc` universale accesibile

Comandă	Descriere
<code>restart nume-imprimanta</code>	Încearcă să reinițializeze daemonul imprimantei (<code>lpd</code>). Un utilizator poate încerca această comandă dacă imprimanta pare OK dar nu se tipărește nimic, chiar dacă <code>lpq</code> afișează lucrări în așteptare în regiunea de derulare. Folosiți <code>a11</code> pentru numele imprimantei dacă doriți să restartați toate imprimantele.
<code>status nume-imprimanta</code>	Afișează starea imprimantei specificate. Dacă utilizatorul nu indică un nume de imprimantă, comanda afișează starea imprimantei denumite <code>lp</code> , adică imprimanta prestabilită.
<code>help nume-comanda</code>	Asigură informații de asistență privind o anumită comandă <code>lpc</code> .
<code>exit</code>	Părăsește <code>lpc</code> (folosită numai în mod interactiv).
<code>quit</code>	Părăsește <code>lpc</code> (folosită numai în mod interactiv).

Tabelul 14-3 prezintă comenzile `lpc` accesibile numai utilizatorului `root`.

Tabelul 14-3 Comenzile `lpc` accesibile numai utilizatorului `root`

Comandă	Descriere
<code>abort nume-imprimanta</code>	Se comportă ca și comanda <code>stop</code> , dar nu permite finalizarea lucrării curente. Când tipărirea este reluată, lucrarea curentă se tipărește din nou.
<code>clean nume-imprimanta</code>	Șterge toate lucrările, inclusiv lucrarea activă, din coada de tipărire.
<code>disable nume-imprimanta</code>	Dezactivează derularea lucrărilor de tipărire pentru o imprimantă specificată. Când derularea este dezactivată, utilizatorii nu mai pot folosi <code>lpr</code> pentru a tipări.
<code>down nume-imprimanta</code> <i>mesaj</i>	Dezactivează derularea și oprește daemonul de tipărire de la tipărirea lucrărilor aflate în coadă (combină acțiunile comenzilor <code>disable</code> și <code>stop</code>). <i>Mesajul</i> va fi afișat când utilizatorii rulează comanda <code>lpq</code> .
<code>enable nume-imprimanta</code>	Activează derularea lucrărilor de tipărire la o imprimantă specificată.
<code>start nume-imprimanta</code>	Activează daemonul de tipărire pentru a putea tipări toate lucrările aflate în directorul de derulare al imprimantei.
<code>stop nume-imprimanta</code>	Așteaptă finalizarea lucrării de tipărire curente și apoi dezactivează daemonul de tipărire pentru a opri tipărirea lucrărilor din directorul de derulare al imprimantei. Când tipărirea este oprită, utilizatorii pot emite în continuare comanda <code>lpr</code> pentru tipărire, dar ieșirea nu apare decât după lansarea daemonului de tipărire cu comanda <code>start</code> .
<code>topq nume-imprimanta</code> <i>identificator-lucrare</i>	Mută lucrarea de tipărire specificată la începutul cozii de tipărire. Dacă folosiți un nume de utilizator în loc de <i>identificator-lucrare</i> , toate lucrările aparținând utilizatorului respectiv sunt deplasate la începutul cozii.
<code>up nume-imprimanta</code>	Inversează acțiunea comenzii <code>down</code> , activează derularea și lansează daemonul de tipărire (combină acțiunile comenzilor <code>enable</code> și <code>start</code>).

/etc/printcap (descrie în secțiunea următoare). Astfel, dacă dispuneți de o imprimantă HP LaserJet denumită HPLJ, directorul de derulare se va numi /var/spool/lpd/hplj. Numele de imprimantă prestabilit este lp și directorul de derulare prestabilit este var/spool/lpd/lp.



Secret

Accidental, o imprimantă poate folosi două nume. De exemplu, o imprimantă HP LaserJet poate fi folosită ca imprimantă care înțelege limbajul de control al imprimantelor (Printer Control Language - PCL) sau PostScript. Puteți folosi nume de imprimantă în fișierul /etc/printcap, descrise în secțiunea următoare.

Chiar și imprimantele de la distanță necesită directoare de derulare pe sistemul dumneavoastră, deoarece lpr mai întâi copiază fișierele în directorul de derulare, după care lpd trimite aceste fișiere către imprimanta de la distanță.

Directorul de derulare al fiecărei imprimante conține două fișiere:

- Fișierul lock, care conține informații despre lucrarea activă și se asigură că un singur lpd tipărește conținutul acestui director de derulare. Procesul lpd manipulează setările de permisiune ale fișierului lock pentru a activa sau dezactiva caracteristica de derulare pentru această imprimantă.

- Fișierul status, care conține un mesaj alcătuit dintr-o singură linie, ce descrie starea imprimantei. Comanda lpq afișează mesajul provenit de la fișierul de stare. Acest mesaj poate fi modificat când se aplică imprimantei comanda lpc down.

Când configurați manual o nouă imprimantă, trebuie să creați un director de derulare pentru imprimantă și să setați în mod corespunzător permisiunea pentru aceasta. Secțiunea „Instalarea și configurarea imprimantei” discută în detaliu etapele de instalare și configurare a imprimantei.

Fișierul /etc/printcap

Fișierul /etc/printcap reprezintă esența mediului de tipărire Linux. Acest fișier text (cu o sintaxă încălcată) descrie caracteristicile diferitelor imprimante. Fișierul trebuie să conțină informații despre fiecare imprimantă care poate fi accesată din sistemul dumneavoastră (inclusiv imprimantele conectate în aceeași rețea ca și calculatorul). Nu se întâmplă nimic rău dacă fișierul /etc/printcap conține descrieri ale unor imprimante de care nu dispuneți. Programele lpr și lpd consultă fișierul /etc/printcap numai pentru a căuta informații despre o anumită imprimantă.

Structura elementară a fișierului /etc/printcap

Fișierul /etc/printcap este modelat după fișierul /etc/termcap, care descrie caracteristicile diferitelor tipuri de terminale. Pentru a înțelege sintaxa intrărilor din /etc/printcap, examinați următoarea intrare demonstrativă pentru o imprimantă HP LaserJet:

```
# HP LaserJet printer
lp|hplj|Laserjet-IRM|HP Laserjet 4M in IRM Division:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:if=/var/spool/lpd/lp/hplj-if.pl:\
:lf=/var/spool/lpd/lp/hplj-log:\
:Sh:
```

Acest exemplu reprezintă o intrare relativ lizibilă din fișierul /etc/printcap. În această intrare sunt menționate următoarele:

- Lucrarea de tipărire este trimisă la o imprimantă denumită /dev/lp1 (câmpul lp).
- Memorează lucrările de tipărire așezate în coadă în directorul /var/spool/lpd/lp (câmpul lp).

- Nu limitează dimensiunea lucrărilor de tipărire (câmpul mx).

- Trece fișierul de tipărire prin filtrul (o comandă care citește din intrarea standard și generează ieșire la ieșirea standard) denumit /var/spool/lpd/lp/hplj-if.pl (câmpul if).

- Consemnează erorile într-un fișier denumit /var/spool/lpd/hplj-log, care trebuie să fie deja existent pentru a permite consemnarea erorilor (câmpul lf).

- Elimină anteturile – o pagină de separație între lucrările de tipărire (câmpul sh).

Rețineți că aveți nevoie de un backslash (\) la sfârșitul fiecărei linii, deoarece se presupune că întreaga intrare reprezintă o singură linie.



Atenție!

Verificați să nu aveți spații suplimentare după caracterul backslash de la sfârșitul unei linii. Dacă după backslash apare un spațiu sau un caracter de tabulare, programele de tipărire (lpr, lpd și lpc) nu vor lua în considerare următoarea linie ca pe o continuare a comenzii. Această eroare poate determina eșuarea procesului de tipărire pentru o imprimantă (chiar dacă intrarea poate arăta excelent când examinați fișierul /etc/printcap).

Ca în cazul majorității fișierelor de configurare în Linux, liniile de comentariu din /etc/printcap încep cu un caracter diez (#).

Nume de imprimantă

Primul câmp al unei intrări (până la primul caracter :) prezintă numele imprimantei și pseudonimele sale. De regulă, fiecare imprimantă are trei nume:

- un nume scurt, cu cel mult patru caractere (precum hplj);
- un nume mai lung, care indică posesorul imprimantei (cum ar fi LaserJet-IRM)
- un nume descriptiv, care trebuie să identifice complet imprimanta pentru utilizatorii din sistemul dumneavoastră (de exemplu, HP LaserJet 4M in IRM Division)



Sugestie

Când tipăriți ceva cu lpr fără a specifica o imprimantă, lucrarea de tipărire respectivă este trimisă automat la imprimanta prestabilită a sistemului, denumită lp. Ca atare, trebuie să scrieți lp ca primul nume al imprimantei prestabilite.

Tipuri de câmpuri

După nume, intrarea aferentă unei imprimante are o secvență de câmpuri. Dacă examinați intrarea prezentată ca exemplu, veți observa separarea câmpurilor prin două puncte (:). Există trei tipuri de câmpuri:

- **Câmpurile șir**, cum este lp (care specifică numele de periferic al imprimantei) sunt specificate prin următoarea sintaxă:

```
:Nume_camp=sir:
```

Puteți încapsula în acest șir caractere speciale prin utilizarea notației cu backslash folosită în limbajul de programare C. Pentru a specifica caracterul de avans linie (utilizabil pentru a extrage o pagină din imprimantă), veți folosi opțiunea \f.

- **Câmpurile numerice**, cum este mx (care specifică dimensiunea fișierelor cu lucrări de tipărire), sunt specificate după cum urmează:

```
:Nume_camp#numar:
```

- **Câmpurile logice**, cum este sh (prin care se elimină tipărirea unei pagini antet de separare a lucrărilor de tipărire consecutive) sunt adevărate când apar în intrarea aferentă imprimantei; în caz contrar, sunt false.

Fiecare câmp are o anumită semnificație. Prin intermediul acestor câmpuri, utilizatorul controlează modul de apariție a ieșirii la imprimantă (de exemplu, dacă fișierul de date trebuie prelucrat de vreun program intermediar anterior ieșirii la imprimantă) și modul de consemnare a erorilor. În secțiunile următoare vom descrie în detaliu câmpurile din fișierul `/etc/printcap`.

Deși conținutul fișierului `/etc/printcap` pare bizar la început, când veți cunoaște formatul de bază și semnificația diferitelor câmpuri, veți fi capabil de a înțelege și edita fișierul cu ușurință.



Sugestie

Secțiunile următoare conțin toate informațiile necesare pentru a înțelege și a adăuga o nouă intrare în fișierul `/etc/printcap`. Dacă trebuie să căutați informații despre câmpurile din fișierul `/etc/printcap`, o puteți face foarte ușor cu următoarea comandă:

```
man printcap
```

Câmpuri din intrările fișierului `/etc/printcap`

Înainte de a învăța despre anumite câmpuri din fișierul `printcap` (mă refer la fișierul `/etc/printcap` cu numele generic `printcap`), consultați tabelul 14-4, care furnizează o listă alfabetică a câmpurilor `printcap`. Coloana *Câmp* afișează numele din două caractere al unui câmp. Coloana *Tip* afișează tipul câmpului (șir, număr sau logic). Coloana *Valoare prestabilită* prezintă valoarea presupusă de programele `lpr` și `lpd` când intrarea din fișierul `printcap` nu specifică acest câmp. Coloana *Descriere* conține o scurtă descriere a câmpului.

Tabelul 14-4 Listă alfabetică a câmpurilor din fișierul `printcap`

Câmp	Tip	Valoare prestabilită	Descriere
af	șir	NULL	Numele fișierului de cont
br	număr	nu are	Rată de baud (se aplică numai dacă câmpul <code>lp</code> specifică un periferic de port serial)
cf	șir	NULL	Nume filtrului de date <code>cifplot</code>
df	șir	NULL	Numele filtrului de date TeX (format DVI)
fc	număr	0	Biții indicator de șters (se aplică numai dacă câmpul <code>lp</code> specifică un periferic de port serial)
ff	șir	\f (salt de pagină)	Șir de trimis pentru un salt de pagină
fo	logic	false	Dacă este specificat, tipărește un salt de pagină când perifericul este deschis
fs	număr	0	Biții indicator de setat (se aplică numai dacă câmpul <code>lp</code> specifică un periferic de port serial)
gf	șir	NULL	Numele filtrului de date grafic
hl	logic	false	Dacă este specificat și <code>sh</code> nu este specificat, tipărește pagina de antet de burst ultima (<i>pagina de antet de burst</i> separă între ele lucrările de tipărire)
ic	logic	false	Dacă este specificat, driverul de imprimantă (identificat prin <code>lp</code>) suportă comanda de control pentru indentarea rezultatului tipărit

Câmp	Tip	Valoare prestabilită	Descriere
if	șir	NULL	Nume al fișierului text, apelat o dată pentru fiecare lucrare de tipărire (îl puteți folosi pentru a interpreta fișierul și pentru a-l converti într-un format adecvat pentru imprimanta dumneavoastră)
lf	șir	/dev/console	Numele fișierului unde sunt consemnate erorile
lo	șir	lock	Numele fișierului de blocare
lp	șir	/dev/lp	Numele perifericului de deschis pentru a trimite ieșire la imprimantă
mx	număr	1000	Dimensiunea maximă a fișierului lucrare de tipărire (în KB); valoarea 0 înseamnă dimensiune nelimitată.
nf	șir	NULL	Numele filtrului de date ditroff (data-independent troff).
of	șir	NULL	Numele programului de filtrare a ieșirii (apelat numai dacă nu este specificat câmpul <code>if</code> și numai o singură dată)
pc	număr	200	Preț pe picior sau pagină (în sutimi de cent)
pl	număr	66	Număr de linii per pagină
pw	număr	132	Lățimea paginii, în număr de caractere
px	număr	0	Lățimea paginii în pixeli (pe orizontală)
py	număr	0	Înălțimea paginii în pixeli (pe verticală)
rf	șir	NULL	Numele filtrului folosit pentru tipărirea fișierelor text stil FORTRAN
rg	șir	NULL	Grup cu restricții; numai membrii grupurilor listate pot accesa imprimanta
rm	șir	NULL	Numele sistemului pentru imprimanta de la distanță
rp	șir	lp	Numele imprimantei de la distanță
rs	logic	false	Dacă este adevărat, tipărește lucrările trimise (de sistemele de la distanță) numai de către utilizatorii cu conturi pe sistemul local
rw	logic	false	Dacă este specificat, deschide perifericul de tipărire pentru citire și scriere
sb	logic	false	Dacă este specificat, tipărește un banner scurt (o singură linie)
sc	logic	false	Dacă este specificat, nu tipărește copii multiple
sd	șir	/var/spool/lpd	Numele directorului de derulare
sf	logic	false	Dacă este specificat, nu tipărește salturi de pagină
sh	logic	false	Dacă este specificat, nu tipărește antetul de pagină de burst
st	șir	status	Numele fișierului de stare
tf	șir	NULL	Numele filtrului de date troff (typesetter)
tr	șir	NULL	Șir trailer de tipărit când coada de tipărire s-a golit
vf	șir	NULL	Numele filtrului de imagine rastru

După cum se vede în tabelul 14-4, puteți specifica peste 35 de câmpuri pentru fiecare intrare printcap. Pentru o intrare obișnuită în fișierul `/etc/printcap`, totuși, trebuie să specificați numai câteva câmpuri. Următoarele nouă sunt printre cele mai folosite câmpuri:

- **if**: un șir care specifică numele filtrului de intrare
- **lf**: un șir care specifică fișierul unde sunt jurnalizate erorile pentru imprimanta respectivă
- **lp**: un șir care specifică numele de periferic al imprimantei (cum ar fi `/dev/lp1`)
- **mx**: un număr care specifică dimensiunea maxim permisă (în blocuri de 1 KB) a unui fișier de tipărire
- **rm**: un șir care specifică numele unui calculator de la distanță
- **rp**: un șir care specifică numele unei imprimante de la distanță
- **sd**: un șir care specifică numele directorului de derulare pentru imprimanta respectivă
- **sf**: o variabilă logică prin care se suprimă saltul de pagină la sfârșitul fiecărei lucrări de tipărire
- **sh**: o variabilă logică prin care se suprimă pagina de antet de burst care apare între două lucrări de tipărire

În secțiunile următoare vom explica în detaliu câteva dintre aceste câmpuri.

Câmpul filtru de intrare

Filtrul de intrare este un script de interpretor al unui program executabil, care citește datele de tipărire de la intrarea standard (`stdin`) și scrie ieșirea la ieșirea standard (`stdout`). Avantajul filtrului de intrare este acela că vi se permite să procesați un fișier text și să-l convertiți într-un format necesar imprimantei dumneavoastră



Daemonul de tipărire apelează filtrul de intrare o dată pentru fiecare lucrare de tipărire. Să presupunem că ați specificat câmpul `if` astfel:

```
:If=/var/spool/lpd/lp/hplj-if.pl:
```

Dacă fișierul `/usr/spool/lp1/hplj-if.pl` este un script Perl, `lpd` invocă filtrul folosind următoarea linie de comandă:

```
/var/spool/lpd/lp/hplj-if.pl [-c] -wwidth -llength -indent -n login h host acct-file
```

Parantezele din jurul opțiunii `-c` arată că este opțională. Opțiunea `-c` este folosită numai dacă utilizatorul ar fi invocat comanda `lpr` cu opțiunea `-l` (ceea ce înseamnă transmiterea nemodificată a tuturor caracterelor la imprimantă). Astfel, dacă ați fi scris filtrul de intrare, argumentele ar fi fost următoarele:

Argument	Descriere
<code>-c</code>	Transmite literal caracterele de control la imprimantă (dacă primul argument este <code>-c</code> , filtrul va copia pur și simplu intrarea standard la ieșirea standard)
<code>-wwidth</code>	Lățimea paginii, în număr de caractere per linie (din câmpul <code>pw</code> din fișierul <code>printcap</code>)
<code>-llength</code>	Înălțimea paginii, în număr de linii per pagină (din câmpul <code>pl</code> din fișierul <code>printcap</code>)
<code>-iindent</code>	Indentarea ieșirii prin tipărirea unui număr de <code>indent</code> spații albe în fața fiecărei linii
<code>-n login</code>	Numele de login al utilizatorului care a trimis lucrarea de tipărire

Argument	Descriere
<code>-h host</code>	Numele sistemului unde utilizatorul a trimis lucrarea de tipărire
<code>acct-file</code>	Numele fișierului de cont, din câmpul <code>af</code> din <code>printcap</code>

Utilizarea filtrelor de intrare va fi descrisă în secțiunea „Evitarea efectului de scară”, în care este descris modul de prevenire al unui asemenea efect la tipărirea de fișiere UNIX pe o imprimantă conectată la un PC, care așteaptă o combinație de caractere retur de car – salt la linie nouă la sfârșitul fiecărei linii.

Perifericul de tipărire

Câmpul periferic de tipărire, `lp`, trebuie să specifice numele de periferic Linux pentru portul la care ați conectat imprimanta. Pentru o imprimantă locală conectată la portul paralel al calculatorului, acest câmp are valoarea `/dev/lp1` (sau `/dev/lp0` pentru unele sisteme mai vechi).

Dacă imprimanta dumneavoastră se găsește pe un port serial, specificați numele de periferic al acestuia - `/dev/ttyS0` pentru COM1 și `/dev/ttyS1` pentru COM2 – drept șirul din câmpul `lp`.

Când setați o intrare în fișierul `printcap` pentru o imprimantă de la distanță, specificați un câmp `lp vid`, după cum urmează:

```
:lp=:
```

Apoi, desigur, trebuie să specificați câmpurile `rm` și `rp`, care specifică numele gazdei și pe acela al imprimantei de la distanță; altfel, `lpd` afișează un mesaj de eroare, deoarece nu poate găsi perifericul de tipărire.

O altă utilizare subtilă a câmpului `lp` constă în setarea acestuia la `/dev/null` – universala *găleată de biți* (nu se întâmplă absolut nimic când datele sunt setate la perifericul `null`). Când încercați o nouă intrare în `printcap` pentru a vedea dacă totul este OK și nu doriți să irosiți hârtia, setați câmpul `lp` din intrarea `printcap` după cum urmează:

```
:lp=/dev/null:
```

Fișierul jurnal

Câmpul fișier jurnal, `lf`, specifică fișierul unde `lpd` trimite mesajele de eroare. Dacă nu specificați nici un fișier, mesajele de eroare ajung la consolă. De regulă, se poate specifica un fișier jurnal în directorul de derulare al imprimantei, după cum urmează:

```
:Lf=/var/spool/lpd/lp/hplj-log:
```

Puteți specifica orice fișier doriți, dar fișierul specificat trebuie să existe. Dacă fișierul nu există, `lpd` nu va consemna erorile.

Eliminarea anteturilor și a saltului de pagină

În terminologia mediului de tipărire Linux, *pagina de antet de burst* sau *pagina banner* desemnează pagina care precede o lucrare de tipărire. În mod prestabilit, `lpd` generează o pagină banner și trimite un caracter salt de pagină pentru a se asigura că fiecare lucrare de tipărire începe pe o pagină nouă.



Referință încrucișată

Probabil că, fiind singurul utilizator al sistemului dumneavoastră Linux, nu veți folosi prea mult paginile de antet. De asemenea, majoritatea programelor integrate de procesare a textelor, precum TeX sau troff (descrie în capitolul 24) deja generează un salt de pagină la finalul tuturor paginilor, deci saltul de pagină trimis de `lpd` generează o pagină albă suplimentară, ceea ce este complet lipsit de sens.

Cele două opțiuni logice `sh` și `sf` dezactivează pagina de antet, respectiv saltul de pagină. Tot ce aveți de făcut este să inserați cele două câmpuri în intrarea aferentă imprimantei din fișierul `printcap`, după cum urmează:

```
:sh:\n
:sf:
```

Stabilirea dimensiunii maxime a unei lucrări de tipărire

Puteți folosi câmpul `mx` pentru a limita dimensiunea unei lucrări de tipărire. Dimensiunea se referă la numărul de baiți din fișierul derulat. Anterior progreselor înregistrate în domeniul graficii și apariției imprimantelor laser, această limită avea un oarecare sens. În prezent, este bine să nu specificați această limită, deoarece fișierele grafice pot avea dimensiuni enorme, chiar dacă ieșirea se limitează la câteva pagini. Pentru a vă asigura că dimensiunea fișierelor de derulare este limitată numai la spațiul disponibil pe disc, specificați zero ca valoare a câmpului `mx`, după cum urmează:

```
:mx#0:
```

Dacă doriți să vă asigurați că fișierele de derulare nu vă ocupă întreg spațiul disponibil pe disc, inserați un fișier denumit `minfree` în directorul de derulare al imprimantei (specificat de câmpul `sd`). În câmpul `minfree`, specificați numărul de blocuri de disc (1 KB per bloc) care trebuie să fie disponibile pentru scrierea de către `lpr` a datelor derulate în directorul de derulare.

Utilizarea mai multor intrări în fișierul `printcap` pentru o singură imprimantă

Pe măsură ce începeți să înțelegeți intrările din fișierul `printcap`, veți vedea că o intrare conține diferite opțiuni de procesare pentru datele care ajung la imprimanta fizică. Dacă o imprimantă acceptă atât PostScript cât și limbajul PCL Hewlett-Packard, puteți avea două intrări separate în fișierul `printcap`: una pentru ieșirea PostScript și alta pentru PCL. Programele care generează PostScript pot folosi numele de imprimantă PostScript, în timp ce programele care vorbesc PCL pot trimite datele de ieșire la imprimanta PCL.

Instalarea și configurarea imprimantei

Când instalați Linux de pe CD-ROM-ul asociat acestei cărți, parcurgând etapele decriptate în capitolul 1, aveți și posibilitatea de a vă instala și configura imprimantele. Ulterior, dacă doriți să adăugați o imprimantă nouă sau să modificați configurația unei imprimante existente, intrați în sistem ca `root` și rulați programul `printtool` sub `X`; lansați `X` și apoi tastați `printtool` într-o fereastră `xterm`.

În secțiunile următoare vom prezenta modul de configurare manuală a unei imprimante. Una dintre primele etape de configurare a unei imprimante constă în pregătirea intrării din fișierul `printcap` aferente imprimantei. O metodă indicată pentru pregătirea unei intrări în fișierul `printcap` este de a încerca o intrare cu caracter demonstrativ.

Un șablon `printcap`

În această secțiune, vom scrie o intrare `printcap` pentru o imprimantă denumită `sablon`. Deoarece directorul `/var/spool/lpd/lp` deja există, am decis să-l folosesc ca director de derulare. Pentru a testa intrarea din `printcap` fără a trimite efectiv date la imprimantă, am folosit `/dev/null` ca nume de periferic de tipărire și am specificat un filtru de intrare care copiază datele de intrare într-un fișier.

Pornind de la aceste presupuneri, am scris următoarea intrare `printcap` și am adăugat-o la fișierul `/etc/printcap`:

```
# Un model printcap demonstrativ
model|model-printer|Imprimanta model care tipareste intr-un fisier:\n
:sd=/var/spool/lpd/lp:\n
:lp=/dev/null:\n
:if=/var/spool/lpd/lp/model-if.tcl:\n
:lf=/var/spool/lpd/lp/model-log:\n
:mx#0:\n
:sh:\n
:sf:
```

Pentru consemnarea erorilor, am creat fișierul `/var/spool/lpd/lp/sample-log` cu următoarele comenzi (după ce am intrat în sistem ca `root`):

```
cd /var/spool/lpd/lp
touch model-log
chmod ug=rw,o=r model-log
```



În continuare, am pregătit filtrul de intrare, `/var/spool/lpd/lp/model-if.tcl`. După cum probabil ați ghicit din numele fișierului, am scris filtrul de intrare în Tcl. Filtrul poate fi scris ca script de interpretor sau în Perl, după preferințe. În realitate, puteți scrie filtrul în C sau C++. Dacă scrieți filtrul în C sau C++, trebuie să efectuați compilarea și legarea pentru a crea fișierul executabil, al cărui nume trebuie specificat în câmpul `if` al intrării `printcap`.

În continuare este prezentat fișierul script Tcl care servește drept filtru de intrare pentru imprimanta model:

```
#!/usr/bin/tcl
#
# Filtru de intrare demonstrativ
#!/usr/bin/tcl
#
# Filtru de intrare demonstrativ
#
# Insereaza-l in directorul de derulare al imprimantei si
# transforma-l in fisier universal executabil
# Filtrul este invocat cu urmatoarea linie de comanda:
#
# <filter> [-c] -wWidth -lLength -iIndent -n Login h Host AcctFile

# Deschide fisierul /tmp/model.out
set outfile [open /tmp/sample.out w]

puts $outfile "-----"
flush $outfile
exec date >@ $outfile
puts $outfile "-----"

# Citeste din stdin si scrie in fisier

while { [gets stdin line] != -1 } {
    puts $outfile $line
}

close $outfile
```

Acest filtru pur și simplu scrie datele de la intrarea standard în fișierul `/tmp/model.out`, adăugând un antet care afișează data și ora tipăririi. Deoarece numele de periferic pentru

intrarea printcap demonstrativă este setat la /dev/null, fișierul /tmp/model.out va fi singura dovadă că tipărirea a dat rezultate.

Rețineți că, într-un filtru de intrare real, scriptul va copia intrarea standard la ieșirea standard, executând toate conversiile necesare (cum ar fi inserția de caractere suplimentare pe fiecare linie sau decalarea unei linii cu un număr determinat de spații albe).

Scriptul devine executabil prin introducerea următoarei comenzi:

```
chmod +x model-if.tcl
```

Pentru a vedea intrarea printcap pentru imprimanta model în acțiune, încercați următoarea comandă:

```
ls -l /var/spool/lpd/lp | lpr /Pmodel
```

Această comandă trimite listingul directorului /usr/spool/lp1 la lpr. Ulterior comanda lpr derulează acest listing pentru tipărire pe imprimanta model (datorită opțiunii -Pmodel). După derularea lucrării de tipărire, daemonul lpd invocă filtrul de intrare pentru imprimanta denumită model. Acest filtru (/var/spool/lpd/lp/ model-if.tcl), la rândul său, copiază listingul directorului în fișierul /tmp/model.out.

Pentru a vedea dacă tipărirea a dat rezultate, verificați conținutul fișierului /tmp/model.out, după cum urmează:

```
cat /tmp/model.out
```

```
-----
Sun May 10 02:53:22 EDT 1998
-----
total 16
-rwxr-xr-x 1 root root 9334 Jan 25 20:58 filter
-rwxr-xr-x 1 root root 189 Jan 25 20:58 general.cfg
-rw-r--r-- 1 root root 19 May 10 02:51 lock
-rwxr-xr-x 1 root root 348 Jan 25 20:58 postscript.cfg
-rwxr-xr-x 1 root root 617 May 10 02:49 model-if.tcl
-rw-rw-r-- 1 root root 0 May 10 02:48 model-log
-rw-rw-r-- 1 root root 29 May 10 02:51 status
-rwxr-xr-x 1 root root 150 Jan 25 20:58 textonly.cfg
```

Așa cum se vede în ieșirea comenzii cat, fișierul /tmp/model.out pare să conțină ieșirea așteptată, adică listingul directorului /var/spool/lpd/lp, cu marca de timp ca antet.

Configurarea imprimantei locale

În secțiunile anterioare v-au fost prezentate diferite aspecte ale mediului de tipărire Linux, inclusiv o intrare printcap demonstrativă și un filtru de intrare. Cu informațiile din secțiunile anterioare, puteți configura o nouă imprimantă pe sistemul dumneavoastră Linux PC. Pentru a configura o imprimantă conectată la portul /dev/lp1 al imprimantei dumneavoastră, parcurgeți următoarele etape:

1. Intrați în sistem ca root, deoarece aveți nevoie de privilegii de superutilizator pentru a parcurge etapele prezentate în continuare.
2. Verificați dacă daemonul de tipărire (lpd) rulează. Folosiți comanda `ps | ax | grep lpd` pentru a vedea dacă este semnalat vreun proces lpd. Dacă ați instalat Linux de pe CD-ROM-ul anexat acestei cărți, daemonul lpd trebuie să ruleze pe sistemul dumneavoastră. Fișierul script /etc/rc.d/rc.M lansează programul lpd în timpul rulării secvenței de boot.

3. Verificați dacă programele de tipărire – lpr, lpq, lprm și lpc – se găsesc în sistemul dumneavoastră. Primele trei programe trebuie să se găsească în directorul /usr/bin, iar lpc se află în directorul /usr/sbin.

4. Creați un director de derulare pentru imprimanta nouă sau pur și simplu folosiți directorul /var/spool/lpd/lp, care există deja în sistemul dumneavoastră.

5. Deschideți fișierul /etc/printcap și adăugați intrarea printcap denumită model, afișată în secțiunea anterioară. Înlocuiți numele (model) cu numele imprimantei dumneavoastră (alegeți un nume scurt pe care vi-l puteți reaminti cu ușurință). Înlocuiți numele de periferic /dev/null cu /dev/lp1 (portul paralel al sistemului dumneavoastră). De asemenea, eliminați câmpul if (puteți adăuga ulterior un filtru de intrare, dacă este necesar). Salvați fișierul /etc/printcap.

6. Verificați dacă imprimanta este corect conectată la calculator. Apăsăți butonul de alimentare și verificați dacă imprimanta este online.

7. Tipăriți o pagină test; încercați comanda `ls -l | lpr -Pnume_imprimanta` (înlocuiți `nume_imprimanta` cu numele imprimantei dumneavoastră). Fișierul tipărit trebuie să apară la imprimantă.

8. Dacă sistemul dumneavoastră Linux PC se află într-o rețea și doriți ca utilizatori din alte sisteme să tipărească la imprimanta dumneavoastră, adăugați numele sistemelor de la distanță la fișierul /etc/hosts.lpd (un fișier text). Pur și simplu adăugați un nume de gazdă pe fiecare linie pentru fiecare gazdă care poate accesa imprimanta conectată la calculatorul dumneavoastră.

Configurarea imprimantei de la distanță

Să presupunem că sistemul dumneavoastră Linux PC se află în rețea. Dacă aveți acces la o imprimantă legată la un alt sistem din rețea, trebuie să puteți tipări pe imprimanta de la distanță respectivă. De asemenea, trebuie să puteți configura o intrare printcap în fișierul dumneavoastră /etc/printcap, înainte de a putea trimite lucrări de tipărire către sistemul de la distanță.

Pentru a configura o intrare printcap pentru o imprimantă de la distanță, parcurgeți următoarele etape:

1. În fișierul /etc/printcap al sistemului local, adăugați o nouă intrare printcap cu un câmp lp vid, setați câmpul rm la numele sistemului de la distanță și setați câmpul rp la numele imprimantei de pe sistemul de la distanță. Pe vechiul meu Sun SPARCStation IPC (deoarece rulează BSD UNIX, folosește aceeași configurație de imprimantă ca și Linux) am setat următoarea intrare printcap pentru a folosi imprimanta model pe sistemul meu Pentium, rulând Linux (numele sistemului este lnbsoft):

```
# O imprimanta de la distanta
lp|lnbsoft-model|Imprimanta de la distanta pe lnbsoft:\
:lp:\
:sd=/usr/spool/lpd:\
:rm=lnbsoft:\
:rp=sample:\
:mx#0:\
:sh:\
:sf:
```

2. În fișierul /etc/hosts.lpd al sistemului de la distanță, adăugați numele sistemului dumneavoastră local. Astfel, dacă doresc să tipăresc pe sistemul lnbsoft din stația mea de lucru Sun denumită lnbsoft, am editat fișierul /etc/hosts.lpd din sistemul lnbsoft și am adăugat următoarea linie: lnbsoft

Fișierul `/etc/hosts` de pe sistemul `lnbsoft`, desigur, trebuie să includă adresa IP a sistemului `lnb` (ați învățat mai multe despre adresele IP în capitolul 16).

După parcurgerea acestor etape, puteți tipări pe sistemul de la distanță. Din stația mea de lucru `Sun`, am tipărit pe imprimanta `model` conectată la `lnbsoft` (deoarece imprimanta `model` a fost astfel configurată).

Probleme de tipărire specifice și soluțiile acestora

De regulă, tipărirea fișierelor text trebuie să se producă imediat. Această secțiune evidențiază unele dintre problemele care pot apărea și sugerează soluții corespunzătoare.

Lucrarea de tipărire a fost trimisă, dar nu există ieșire

Această problemă este printre cele mai frustrante. Ați trimis o lucrare de tipărire la imprimantă prin intermediul `lpr` (verificând dacă ați folosit opțiunea `-P` pentru a denumi imprimanta) și nu s-au semnalat mesaje de eroare, dar din imprimantă nu iese nimic. Încercați etapele următoare pentru a identifica motivele:

1. Verificați imprimanta fizică. Asigurați-vă că este conectată la calculator, este alimentată cu curent și este online.
2. Verificați dacă imprimanta este conectată la perifericul specificat în câmpul `lp` al fișierului `printcap` aferent imprimantei. De asemenea, verificați ca perifericul să nu fie `/dev/null`.
3. Rulați `lpq` și verificați dacă coada de tipărire afișează lucrarea trimisă.
4. Tastați `lpc stat` pentru a examina starea imprimantelor și verificați dacă sunt activate tipărirea și coada de tipărire pentru imprimanta respectivă.
5. Verificați dacă daemonul de tipărire (`lpd`) rulează. Dacă nu, tastați `lpd` pentru a lansa daemonul.
6. Verificați dacă toate filtrele de intrare specificate în câmpul `if` din intrarea `printcap` sunt prezente în directorul adecvat și au permisiune de execuție. Căutați eventualele mesaje de eroare din jurnalul de sistem folosind următoarea comandă:

```
tail /var/log/messages
```

Dacă mesajele includ intrări în care se spune `cannot execv nume_filtru`, puteți fi sigur că filtrul nu există sau că nu are setările de permisiune corect configurate. Dacă filtrul nu are permisiune de execuție, tastați `chmod +x nume_filtru` pentru a modifica setările de permisiune.

7. Verificați fișierul pe care doriți să-l tipăriți. Unele imprimante PostScript pur și simplu ignoră fișierele text simplu. (Există filtre care convertesc textul simplu în PostScript. Poate este necesar să instalați un asemenea filtru.)

Probleme cu tipărirea pe imprimante de la distanță

Dacă trimiteți o lucrare de tipărire la o imprimantă de la distanță, iar rezultatul nu apare la imprimanta respectivă, examinați conținutul următoarelor câmpuri din fișierul `/etc/printcap` inclus în sistemul de la distanță:

- Verificați dacă câmpul `rg` specifică grupurile care au permisiunea de a tipări. Dacă numele dumneavoastră nu se află în grupul respectiv, lucrarea de tipărire nu va fi procesată.
- Verificați dacă apare câmpul `rs`. În caz afirmativ, trebuie să aveți un cont pe sistemul îndepărtat pentru a putea tipări pe imprimanta de la distanță.

Evitarea efectului de scară

În UNIX, fiecare linie dintr-un fișier text se încheie cu un caracter *linie nouă* (salt la linie nouă). La mașinile de scris, apăsarea tastei de salt la linie nouă determină avansul hârtiei la rândul următor, dar nu aduce căruciorul la marginea hârtiei. Pentru ca aceasta să se întâmple, trebuie să apăsați tasta retur de car. Dacă ați fi tipărit un fișier text UNIX pe un periferic similar unei mașini de scris, rezultatul ar fi fost următorul:

```
Prima linie
      A doua linie
            A treia linie
                  A patra linie
```

După cum se poate vedea, liniile de text arată ca o scară, de unde și originea termenului *efect de scară*.

MS-DOS și multe imprimante pentru PC se comportă ca mașinile de scris: fiecare linie de text trebuie urmată de un retur de car și un salt la linie nouă pentru a avansa în mod corespunzător la următoarea linie de pe hârtie. Când trimiteți un fișier text UNIX la o imprimantă care așteaptă un retur de car și un salt la linie nouă pentru fiecare linie, iar ieșirea rezultantă a afișat efectul de scară.



Puteți evita efectul de scară în două moduri, care necesită ambele un filtru de intrare:

- Unele imprimante pot fi setate pentru a trata un caracter salt la linie nouă (LF) sub forma unui retur de car, urmat de un salt la linie nouă (CR+LF). De exemplu, familia de imprimante HP LaserJet și Deskjet, de exemplu, pot fi programate pentru a trata CR sub forma CR+LF prin trimiterea secvenței `ESC &k2G` (tasta Escape urmată de secvența `&k2G`). Puteți folosi un filtru de intrare pentru a trimite această secvență la imprimantă înainte de începutul fișierului.

- Dacă imprimanta nu poate fi programată pentru a „vedea” LF sub forma CR+LF, puteți scrie un filtru de intrare care convertește fiecare LF de la sfârșitul unei linii într-un CR, urmat de un LF.

În continuare este prezentat un filtru de intrare, care trimite comanda specială de imprimantă către imprimantele LaserJet pentru a activa interpretarea LF sub forma CR+LF:

```
#!/bin/bash
#Filtru de intrare pentru tratarea LF ca CR+LF de catre
#imprimantele LaserJet si DeskJet
```

```
#Trimite comanda pentru interpretarea LF sub forma CR+LF
echo -ne \\033\k2G
```

```
#Urmatoarea comanda trimite stdin la stdout
cat
```

```
#Urmatoarea comanda trimite un salt la linie noua la sfarsitul
#fisierului
echo -ne \\f
```


Dacă salvați acest script în fișierul `/var/spool/lpd/lp/hplj-if.sh` (și îl faceți executabil cu comanda `chmod +x`), puteți folosi filtrul prin inserția următorului câmp în intrarea `printcap` a imprimantei respective:

```
:if=/var/spool/lpd/lp/hplj-if.sh:
```

Filtrarea unei lucrări de tipărire destinate unei imprimante de la distanță

După ce folosiți `lpr` pentru a derula o lucrare de tipărire destinată unei imprimante de la distanță (care dispune de o intrare `printcap` în sistemul dumneavoastră), procesul `lpd` al sistemului dumneavoastră transferă imediat fișierele asociate lucrării de tipărire procesului `lpd` al sistemului de la distanță. Aceasta înseamnă că, chiar dacă specificați un filtru de intrare în `printcap` pentru imprimanta de la distanță, procesul `lpd` al sistemului dumneavoastră nu va aplica nici un filtru. În schimb, procesul `lpd` al sistemului de la distanță aplică toate filtrele specificate în fișierul `/etc/printcap` al sistemului respectiv.

Dacă nu aveți acces la fișierul `/etc/printcap` al sistemului de la distanță pentru a adăuga un filtru de intrare, puteți folosi un truc în sistemul dumneavoastră pentru a aplica un filtru de intrare la lucrarea de tipărire, înainte de a ajunge la imprimanta de la distanță.

Creați o intrare `printcap` demonstrativă, care să specifice filtrul de intrare pe care doriți să-l aplicați. În acest filtru de intrare, faceți ce vreți cu datele și apoi redirecționați-le spre imprimanta de la distanță. Dacă doriți să adăugați secvența de programare astfel ca o imprimantă HP de la distanță să interpreteze fiecare LF sub forma CR+LF, folosiți un script filtru similar cu următorul:

```
#!/bin/bash
{
# Retineti ca \033 este Esc in octal
echo -ne \\033\\&k2G
cat
echo -ne \\f
} | lpr -Php-remote -h -l
```

Acest exemplu presupune că fișierul `/etc/printcap` al sistemului dumneavoastră se referă la imprimanta de la distanță sub numele `hp-remote`.

Evitarea fișierelor grafice trunchiate

Fișierele grafice pot avea dimensiuni mari. Dacă aveți probleme cu tipărirea fișierelor grafice datorită trunchierii acestora, verificați intrarea `printcap` pentru imprimanta respectivă. Dacă intrarea respectivă nu are un câmp `mx`, dimensiunea maximă a unui fișier din directorul de derulare este de 1000 de blocuri. Orice fișier de o dimensiune peste 1000 de blocuri este trunchiat în mod automat.

Pentru a evita trunchierea, adăugați următorul câmp `mx` la intrarea `printcap`:

```
mx#0
```

Această intrare permite o dimensiune nelimitată a fișierelor de derulare.

Comanda `lpr -i` nu determină decalarea ieșirii

Opțiunea `-i` a comenzii `lpr` are scopul de a decala ieșirea cu un număr specificat de spații albe. Pentru ca această opțiune să fie funcțională, trebuie furnizat un filtru de intrare, deoarece `lpr` se rezumă la transmiterea opțiunii `-i` către filtrul de intrare. Practic, ieșirea va fi decalată numai dacă filtrul de intrare manipulează opțiunea `-i`.

Tipărirea fișierelor PostScript

Dacă imprimanta dumneavoastră este PostScript, puteți tipări un fișier PostScript prin simpla inserție a acestuia în directorul de derulare al imprimantei respective. Dacă nu dispuneți de o asemenea imprimantă, puteți folosi programul Ghostscript pentru a tipări fișiere PostScript pe mai multe tipuri de imprimante. De fapt, puteți previzualiza fișiere PostScript folosind programul Ghostview, care folosește Ghostscript pentru a genera date de ieșire adecvate pentru afișarea într-o fereastră X.



CD

Programul Ghostscript este inclus pe CD-ROM-ul anexat acestei cărți. Opțiunea de instalare a acestui program apare la instalarea sistemului Linux, prin parcurgerea etapelor prezentate în capitolul 1.

Tabelul 14-5 prezintă perifericele de ieșire suportate de versiunea (3.33) a programului Ghostscript existentă pe CD-ROM-ul anexat. După cum se poate vedea, între perifericele de ieșire sunt incluse echipamente hardware ca ecranul de afișare sau imprimanta, dar și multe formate frecvent folosite de fișiere imagine, precum TIFF sau PCX.

Tabelul 14-5 Periferice compatibile Ghostscript

Nume periferic	Descrierea perifericului de ieșire
bit	Bitmap monocrom
bitcmymk	Bitmap cu culori în format CMYK (Cyan-Magenta-Yellow-Black)
bitrgb	Bitmap cu culori în format RGB (REd-Green-Blue)
bj10e	Imprimantă Canon BubbleJet BJ10e
bj200	Imprimantă Canon BubbleJet BJ200
bjc600	Imprimante Canon BubbleJet BJC-600, BJC-4000 și BJC-70
cdeskjet	Imprimantă color HP DeskJet 500C în mod 1 bit-per-pixel
cdj500	Imprimantă HP DeskJet 500C (similar cu <code>cdjcolor</code>)
cdj550	Imprimante HP DeskJet 550C, 560C și 660Cse
cdjcolor	Imprimantă HP DeskJet în mod 24 de biți per pixel și dithering (Floyd-Steinberg) al culorii de înaltă calitate (util și pentru imprimantele HP DeskJet 540C și Citizen ProJet 11c)
cdjmono	Imprimantă HP DeskJet 500C în mod alb-negru (de asemenea recomandat pentru DeskJet 510, 520 și 540C în mod alb-negru)
cp50	Imprimantă color Mitsubishi CP50
deskjet	Imprimante HP DeskJet și DeskJet Plus
dfaxhigh	Format software DigiFAX (de la DigiBoard, Inc.)
dfaxlow	Format DigiFAX pentru rezoluție joasă (normală)
djet500	Imprimantă HP DeskJet 500
djet500C	Imprimantă HP DeskJet 500C (nu este compatibil cu 550C sau 560C)
dnj650c	Imprimantă HP DeskJet 650C

(continuare)

Tabelul 14.5 Continuare

Nume periferic	Descrierea perifericului de ieșire
eps9high	Imprimantă compatibilă Epson pe 9 pini în mod triplă rezoluție (linii intercalate)
eps9mid	Imprimantă compatibilă Epson pe 9 pini în mod rezoluție intermediară (linii intercalate)
epson	Imprimante matriceale compatibile Epson (pe 9 sau 24 pini)
epsonc	Imprimante color Epson LQ-2550 și Fujitsu 3400/2400/1200
faxg3	Fișier FAX Group 3 standard ITU (International Telecommunications Union), cu EOL-uri dar fără antet sau EOD
faxg32d	Fișier FAX Group 3 bidimensional standard ITU, cu EOL-uri dar fără antet sau EOD
faxg4	Fișier FAX Group 4 standard ITU, cu EOL-uri dar fără antet sau EOD
ibmpro	Imprimantă IBM Proprinter pe 9 pini
jetp3852	Imprimantă color cu jet de cerneală IBM Jetprinter (Model #3852)
laserjet	Imprimantă HP LaserJet
ljet2p	Imprimantă HP LaserJet IID/IIP/III cu comprimare TIFF
ljet3	Imprimantă HP LaserJet III cu comprimare Delta Row
ljet3d	Imprimantă HP LaserJet IID cu posibilitate de tipărire duplex (față-verso)
ljet4	Imprimantă HP LaserJet 4 (rezoluție prestabilită de 600 puncte per inch)
ljetplus	Imprimantă HP LaserJet Plus
pbm	Format Plain Portable Bitmap
pbmraw	Format brut Plain Portable Bitmap
pcx16	Format PCX pentru plăcile video mai vechi EGA/VGA pe 16 culori (format planar pe 4 biți)
pcx24b	Format PCX pentru plăcile video pe 24 de biți (3 planuri pe 8 biți)
pcx256	Format PCX pentru 256 de culori (format chunky pe 8 biți)
pcxgray	Format PCX pe 8 biți pentru nunațe de gri
pcxmono	Format PCX monocrom (1 bit per pixel)
pgm	Format Plain Portable Graymap
pgmraw	Format brut Plain Portable Graymap
pj	Imprimantă HP PaintJet XL
pjxl	Imprimantă color HP PaintJet XL
pjxl300	Imprimantă color HP PaintJet XL300 (compatibil și cu PaintJet 1200C)
ppm	Format Plain Portable Pixmap
ppmraw	Format brut Plain Portable Pixmap
tiffcrle	Format Tagged Image File Format (TIFF) care folosește formatul FAX standard ITU monodimensional Huffman Run Length Encoded (RLE) (în speță FAX Group 3 fără caractere de sfârșit de linie)

Nume periferic	Descrierea perifericului de ieșire
tiffg3	Fișier TIFF care folosește formatul FAX Group 3 standard ITU (FAX Group 3 cu caractere de sfârșit de linie)
tiffg32d	Fișier TIFF care folosește formatul FAX Group 3 bidimensional standard ITU
tiffg4	Fișier TIFF care folosește formatul FAX Group 4 standard ITU
tiff1zw	Format TIFF care folosește comprimarea Lempel-Ziv Welch (LZW)
tiffpack	Format TIFF care folosește comprimarea PackBits
x11	Driver de fereastră X (pentru afișarea datelor de ieșire în fereastră)
x11alpha	Driver de fereastră X cu funcție alfa pe 4 biți (transparentă)
x11cmyk	Driver de fereastră X care acceptă culori în format CMYK (Cyan-Magenta-Yellow-Black)
x11mono	Driver de fereastră X monocrom (alb și negru)

Programul Ghostscript este lansat folosind comanda `gs`. Pentru a specifica un nume de dispozitiv (periferic) se folosește opțiunea `-sDEVICE=nume_periferic`. Pentru a interpreta fișierul PostScript `testfile.ps` și a genera date de ieșire pentru o imprimantă HP LaserJet 4, de exemplu, veți introduce următoarea comandă:

```
gs -sDEVICE=ljet4 testfile.ps
```

Pentru a tipări fișiere PostScript, veți defini o intrare `printcap` pentru o imprimantă PostScript (de regulă, cu un nume precum `ps`) și veți specifica un filtru de intrare pentru intrarea respectivă. Apoi, în cazul filtrului de intrare, veți folosi `gs` pentru a crea ieșirea pentru imprimanta dumneavoastră. Dacă aveți o imprimantă HP DeskJet 550C, veți insera următoarea linie în interiorul unui script de interpretor care servește drept filtru de intrare pentru intrarea `printcap` aferentă imprimantei menționate:

```
/usr/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=cdj550 -sOutputFile
```

Rezumat

Tipărirea în Linux este simplă, cu condiția ca mediul de tipărire să fie corect configurat. Trebuie să învățați un mic set de comenzi pentru a trimite o lucrare de tipărire și a verifica starea lucrărilor de tipărire suspendate. Acest capitol explică mediul de tipărire Linux din punctul de vedere al utilizatorului, după care explică modul de configurare a acestuia de către administratorul de sistem.

Prin lectura acestui capitol, ați învățat următoarele:

- În Linux, numele de periferic al imprimantei depinde de portul la care se conectează aceasta. Pentru o imprimantă obișnuită conectată la portul paralel, numele de periferic este `/dev/lp0` sau `/dev/lp1`, în funcție de adresa I/O a portului paralel. Dacă imprimanta este conectată la portul serial, numele de periferic este `/dev/ttyS0` (pentru COM1), respectiv `/dev/ttyS1` (pentru COM2).

- Ca utilizator, tipăriți cu comanda `lpr`, verificați starea cozii de tipărire cu `lpq`, anulați lucrările de tipărire cu `lprm` și verificați starea imprimantei cu comanda `lpc status`.

- ▶ În fundal, comanda `lpr` copiază fișierul de tipărit într-un director de stocare denumit *director de derulare*. Un proces de fundal denumit `lpd` (daemonul de tipărire) trimite fișierul de tipărire din directorul respectiv la imprimantă.
- ▶ Pentru a configura mediul de tipărire în Linux, trebuie ca imprimanta dumneavoastră să fie definită în fișierul `/etc/printcap`. În esență, acest fișier specifică operațiile care trebuie efectuate la trimiterea unei lucrări de tipărire la imprimantă. Câmpurile din fișierul `etc/printcap` controlează diferitele aspecte ale tipăririi, de la dimensiunea paginii la tipărirea unei pagini de antet înaintea fiecărei lucrări de tipărire.
- ▶ În fișierul `/etc/printcap`, puteți specifica un filtru de intrare (proces care citește de la intrarea standard și scrie la ieșirea standard) prin care trebuie să treacă toate lucrările de tipărire. Filtrul de intrare poate trimite comenzi speciale la imprimantă sau poate modifica fișierul de tipărire, dacă este necesar.
- ▶ Efectul de scară se referă la aspectul datelor tipărite la tipărirea unui fișier text UNIX (unde fiecare linie se încheie cu un singur caracter de salt la linie nouă) pe o imprimantă PC, care se așteaptă ca fiecare linie să se încheie cu o pereche retur de car-salt la linie nouă. Pentru a evita efectul de scară se poate folosi un filtru de intrare.
- ▶ Puteți configura o intrare de imprimantă în fișierul `/etc/printcap`, pentru ca tipărirea să aibă loc pe un alt sistem din rețea.
- ▶ Puteți folosi programul Ghostscript (inclus pe CD-ROM-ul anexat) pentru a vizualiza și tipări fișierele PostScript. Multe pachete software disponibile gratuit includ documentație în format PostScript.

Capitolul 15

Modemuri

În acest capitol

- ▶ Veți folosi numele perifericelor de port serial în Linux
- ▶ Veți înțelege comunicațiile seriale
- ▶ Veți examina standardul RS-232C
- ▶ Veți înțelege diferitele tipuri de standarde pentru modemuri, precum V.34 și V.90
- ▶ Veți înțelege setul de comenzi standard Hayes AT folosit de modemuri
- ▶ Veți apela alte sisteme cu un modem în Linux
- ▶ Veți configura Linux pentru recepția apelurilor
- ▶ Utilizarea plăcilor I/O seriale multiport în Linux

Dacă v-ați instalat Linux pentru a-l utiliza pe calculatorul dumneavoastră personal, puteți folosi un modem pentru a apela prin telefon sistemul dumneavoastră de la birou sau un sistem de buletine electronice (BBS). Dată fiind creșterea popularității Internetului, vă mai puteți conecta la Internet prin intermediul modemului. Pentru a folosi un modem, trebuie să învățați modul în care Linux manipulează *porturile seriale* ale calculatorului – porturile prin care modemul colaborează cu calculatorul. De asemenea, trebuie să vă procurați cabluri pentru a vă conecta modemul la calculator, să selectați perifericul serial adecvat pentru a-l utiliza în Linux și să setați fișiere de configurare care controlează parametrii de comunicare ai portului serial.

Terminalele – periferice care dispun de un ecran de afișare și o tastatură – se conectează de asemenea la un calculator personal prin intermediul porturilor seriale. În acest sens, terminalele sunt similare cu modemurile. Terminalele sunt ideale pentru operații simple de introducere a datelor și sunt frecvent folosite în sistemele din punctele de vânzare. Linux mai include suport și pentru plăcile seriale multiport, care vă permit să conectați mai multe terminale sau modemuri la calculatorul dumneavoastră care rulează Linux.

Acest capitol prezintă modul de conectare, configurare și utilizare a modemurilor și a terminalelor. O mare parte din informații se aplică tuturor perifericelor conectate la portul serial, dar unele informații sunt specifice modemurilor. Acest capitol se concentrează mai ales asupra modemurilor, deoarece acestea sunt mai populare decât terminalele în lumea utilizatorilor Linux.

Calculatorul personal și porturile seriale

Dacă ați mai folosit programe de comunicații, precum Procomm sau Crosstalk, sub MS-DOS sau Windows, înseamnă că ați mai folosit și portul serial al calculatorului dumneavoastră. Aceste programe de comunicații transferă baiți de date de la calculator către modem prin intermediul portului serial. Portul serial are această denumire deoarece fiecare bait de date este trimis *serial* – bit cu bit. Comunicațiile seriale de date dispun de propria terminologie. Pentru a înțelege modul de configurare a unui port serial, trebuie să cunoașteți terminologia acestuia.

UART

Un cip denumit *transmițător/receptor asincron universal* (*Universal Asynchronous Receiver/Transmitter* – UART), care stă la baza tuturor echipamentelor de comunicații seriale, se ocupă de conversia fiecărui bait într-un șir de 1 și 0. Șirul respectiv este apoi trimis prin intermediul mediului de comunicare (de exemplu, linia telefonică). La capătul receptor, un alt UART reconstituie baiții de date pornind de la șirul de 1 și 0.

Placa de bază a calculatorului original IBM-PC nu includea nici un fel de facilități de comunicații. În schimb, funcția de comunicații seriale a fost asigurată prin intermediul unei plăci adaptor seriale separate (sau *placă serială*) denumită adaptor asincron de comunicații IBM (IBM Asynchronous Communications Adapter). Această placă serială a folosit cipul INS8250 UART al firmei National Semiconductor.

Ulterior, PC-urile au început să includă echipamente hardware pentru comunicații seriale pe placa de bază, dar cipul 8250 sau un alt compatibil UART au mai fost folosite multă vreme după aceea. Cu toate acestea, a fost introdusă o versiune îmbunătățită a lui 8250 (și anume 16450), dar îmbunătățirile se regăseau în detaliile de fabricație ale cipului, nu în funcționalitatea de bază a acestuia.

O problemă a cipurilor UART 8250 și 16450 este inexistența unei modalități de bufferizare (memorare) a caracterelor receptate. Dacă calculatorul nu poate „ține pasul” cu fluxul de caractere, este posibil ca unele date să se piardă. La rate de transfer ridicate (peste 9600 biți pe secundă), calculatorul poate avea probleme la sosirea caracterelor, mai ales cu sisteme de operare care mențin în permanență calculatorul ocupat (cum este Microsoft Windows).

Soluția a constat în inserția unui buffer FIFO pe cipul UART, astfel că calculatorul își mai poate permite să piardă ritmul, fără a pierde însă caracterele care sosesc. Noile cipuri UART 16550A dispun de buffere de transmisie și de recepție, fiecare capabil de a stoca maximum 16 caractere. Cipul UART 16550A este compatibil cu vechile 8250 și 16450, dar poate suporta rate de transfer superioare, deoarece bufferele încorporate pot stoca caracterele care vin și care pleacă direct pe UART.

Ca notă istorică, cipul original UART 16550 de la National Semiconductor dispunea de bufferele FIFO, dar circuitele FIFO aveau unele defecte. Cipul 16550A (și versiunile sale ulterioare) a remediat problema cipului 16550 original. Până în acest moment, totuși, numele 16550 este frecvent folosit într-o manieră generică, pentru a desemna cipurile UART care dispun de buffere de transmisie și recepție încorporate pe placă. Nu trebuie să vă faceți probleme vizavi de diferențele dintre 16550 și 16550A, cu excepția cipului original 16550 de la National Semiconductor; aceste cipuri sunt marcate NS16550 (fără A).



Linux include suport pentru porturile seriale I/O care folosesc un UART 8250, 16450, 16550A sau compatibile. Aveți nevoie de un UART 16550A pentru a putea ține pasul cu modemurile rapide actuale, care pot transfera date cu viteze de până la 56600 de biți pe secundă. Majoritatea calculatoarelor Pentium PC noi sunt deja dotate cu cipuri UART 16550A. Dacă vă cumpărați un calculator nou pentru a rula Linux, trebuie să vă asigurați că interfața serială folosește cipuri UART 16550A. În prezent, totuși, au apărut UART-uri mai noi, cum este 16554, care conține un buffer pe 64 de baiți. Și aceste cipuri UART funcționează sub Linux.

Parametri de comunicație

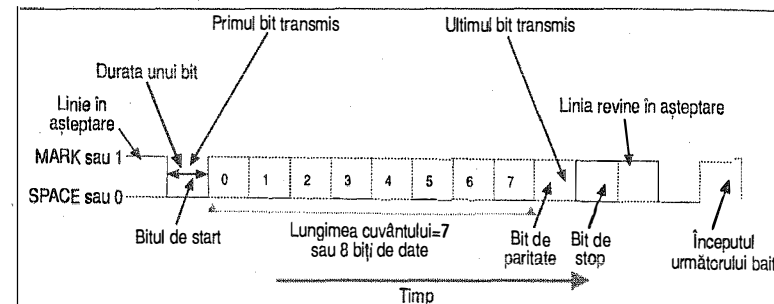
În comunicațiile seriale, mediul de transmisie (cum este linia telefonică) este menținut la 1 logic când este în așteptare (1 este reprezentat prin prezența unui semnal pe linie). În acest caz,

se spune că linia este în stare *mark*. Pe de altă parte, când linia se află la 0 logic, se spune că este în stare *space*. Astfel, stările de 1 și 0 logic sunt de asemenea desemnate sub numele de MARK, respectiv SPACE.

O secvență de 1 și 0 alcătuiește un caracter pe 8 biți, așa cum se vede în figura 15.1.

Figura 15-1

Formatul unui caracter unic în comunicațiile seriale



Așa cum se vede în figura 15.1, o modificare a condiției liniei de la MARK la SPACE arată începutul unui caracter. Această modificare a condiției liniei este cunoscută sub numele de bit de start. După bitul de start urmează o secvență de biți care reprezintă caracterul și un bit cunoscut sub numele de bit de paritate (PARITY). În final, linia revine la condiția de așteptare MARK, care reprezintă bitul de stop și indică sfârșitul caracterului curent. Numărul de biți folosit pentru reprezentarea caracterului este cunoscut sub numele de *lungimea cuvântului* și este, de regulă, șapte sau opt.

Bitul de paritate este folosit pentru a efectua o rudimentară detecție de erori. La selectarea unei *parități pare*, de exemplu, bitul de paritate este setat astfel încât numărul total de 1 din cuvântul curent este 7 (logica este similară pentru paritatea impară). La capătul receptor, paritatea este recalculată și comparată cu bitul de paritate primit. Dacă există o neconcordanță, receptorul declară o eroare de paritate. O problemă care apare la detecția erorilor folosind verificarea parității este că se pot detecta numai erorile care afectează un singur bit. Secvența de biți 0100 0001 0 (codul ASCII pentru litera A), de exemplu, transmisă cu o lungime de cuvânt de opt biți și paritate pară, se poate modifica (datorită zgomotului din linia telefonică) în 0100 0111 0 (ASCII G). Cu toate acestea, receptorul nu va sesiza eroarea, deoarece paritatea rămâne pară. Din acest motiv, detecția erorilor prin metoda parității este rareori folosită în prezent.

În comunicațiile seriale, transmițătorul și receptorul trebuie să dispună ambii de date privind durata fiecărui bit; în caz contrar, nu pot detecta biții în mod corect. Durata fiecărui bit este determinată de ceasurile de date de la receptor și de la transmițător. Observați că, deși ceasurile de la receptor și transmițător trebuie să aibă aceeași frecvență, nu este necesară sincronizarea lor. Bitul de start indică receptorului că urmează un nou caracter; receptorul începe apoi să detecteze biții până când vede bitul de stop.

O anumită condiție a liniei este uneori folosită pentru a obține atenția receptorului. Starea normală a liniei este MARK (1), iar începutul unui caracter este indicat de starea SPACE (2). Dacă linia rămâne în starea SPACE pentru o perioadă de timp mai lungă decât cea necesară pentru a primi toți biții unui caracter, se presupune apariția unei întreruperi (BREAK). Când receptorul vede starea BREAK, se poate pregăti din nou pentru a primi caractere.

Selecția frecvenței de ceas depinde de *rata de baud* (sau pur și simplu *baud*), care se referă la numărul de modificări de stare a liniei într-o secundă. De regulă, echipamentul serial I/O

folosește o frecvență de ceas egală cu de 16 ori rata de baud, deci linia este eșantionată suficient de frecvent pentru a detecta un bit în condiții de fiabilitate.



Observație

Rata de transfer a datelor este exprimată în *biți pe secundă (bps)*. La începutul utilizării modemurilor, rata de transfer a datelor era similară cu rata de baud – numărul de modificări ale stării liniei într-o secundă – deoarece fiecare stare de linie transporta un singur bit de informație. În prezent, modemurile folosesc tehnologii mai avansate, care permit trimiterea mai multor biți la fiecare modificare a stării liniei. Cu alte cuvinte, puteți obține o valoare în bps mare cu o rată de baud relativ redusă. Chiar și cele mai rapide modemuri folosesc o rată de baud de 2400, chiar dacă numărul de biți pe secundă poate ajunge la 56600.



Secret

În consecință, termenul *baud* și cel de *biți pe secundă* nu sunt echivalente. Majoritatea utilizatorilor, însă, ignoră definiția exactă a *ratei de baud* și folosesc termenul de *baud* pentru a se referi la biți pe secundă. Chiar dacă folosim bps când ne referim la viteze de transfer al datelor, veți întâlni setări Linux care folosesc termenul *rată de baud* când se referă la biți pe secundă.

IRQ-uri și adrese de porturi I/O seriale

Un calculator personal dispune de două porturi seriale, denumite COM1 și COM2 în jargonul DOS. Un PC mai conține, de asemenea, două porturi seriale: COM3 și COM4. Datorită acestor nume de porturi, porturile seriale mai sunt denumite și porturi COM.



Ca și alte periferice, porturile seriale necesită numere de cerere de întrerupere (IRQ) și adrese de port I/O. Cele patru porturi COM folosesc două IRQ-uri: 3 și 4. Tabelul 15-1 prezintă IRQ-urile și adresele de port I/O atribuite celor patru porturi seriale.

Tabelul 15-1 Adrese I/O și IRQ-uri atribuite porturilor seriale

Port	IRQ	Adresă I/O
COM1	4	0x3f8
COM2	3	0x2f8
COM3	4	0x3e8
COM4	3	0x2e8

Numele de periferic ale porturilor seriale folosite în Linux

Ca și alte periferice în Linux, porturile seriale sunt reprezentate prin fișiere de periferic situate în directorul `/dev`. Fiecare port serial are două nume: unul pentru conexiunile de intrare (incoming) și altul pentru conexiunile de ieșire (outgoing). Tabelul 15-2 prezintă numele de periferic pentru porturile seriale de intrare și de ieșire, inclusiv numerele majore și minore de periferic.

Tabelul 15-2 Nume de periferic pentru porturile seriale de intrare și de ieșire

Port COM	Nume periferic	Număr major de periferic	Număr minor de periferic
COM1 de intrare	<code>/dev/tty/S0</code>	4	64
COM2 de intrare	<code>/dev/tty/S1</code>	4	65

Port COM	Nume periferic	Număr major de periferic	Număr minor de periferic
COM3 de intrare	<code>/dev/tty/S2</code>	4	66
COM4 de intrare	<code>/dev/tty/S3</code>	4	67
COM1 de ieșire	<code>/dev/cua0</code>	5	64
COM2 de ieșire	<code>/dev/cua1</code>	5	65
COM3 de ieșire	<code>/dev/cua2</code>	5	66
COM4 de ieșire	<code>/dev/cua3</code>	5	67

Dacă ați instalat Linux din CD-ROM-ul anexat acestei cărți, toate aceste periferice trebuie să se găsească deja în sistemul dumneavoastră. Dacă examinați directorul `/dev`, veți descoperi perifericele `/dev/ttyS*` și `/dev/cua*`, unde `*` poate fi înlocuit cu 0, 1, 2 și 3.



Observație

În trecut, Linux folosea perifericele `/dev/ttyS*` pentru apelarea din exterior, iar perifericele `/dev/tty/cua*` pentru apelarea în exterior. Versiunea curentă a nucleului Linux nu face diferență între cele două tipuri de periferice. Sfatul meu este să folosiți numele de periferic `/dev/ttyS*` de fiecare dată când trebuie să faceți referire la periferice seriale.

Modemuri

Termenul de *modem* provine de la *modulator/demodulator* – un dispozitiv care convertește semnalele digitale (alcătuite din succesiuni de 1 și 0) în semnale analogice cu variație continuă, care pot fi transmise prin linii telefonice și unde radio. Astfel, modemul este intermediarul între lumea digitală a calculatoarelor și lumea analogică a telefoanelor.

În interiorul calculatoarelor, 1 și 0 sunt reprezentate prin nivele de tensiune, dar semnalele transportate prin intermediul liniilor telefonice sunt tonuri de diferite frecvențe. Modemul se găsește între cipul UART și linia telefonică și face posibilă comunicarea datelor prin intermediul liniilor telefonice. Modemul convertește informația (într-un sens sau în celălalt) între reprezentarea tensiune/fără tensiune a circuitelor digitale și semnalele analogice adecvate pentru transmisiunea prin linii telefonice.

Comunicația dintre calculatorul personal și modem se conformează standardului RS-232C (numit frecvent și RS-232, fără C). Protocolul de comunicații între două modemuri respectă, de asemenea, unul dintre numeroasele standarde internaționale referitoare la modemuri. În următoarele secțiuni vom descrie pe scurt aceste standarde.

Standardul RS-232C

Standardul RS-232C, creat de Electrical Industry Association (EIA), specifică o metodă recomandată de transfer de informații între modem și echipamentul de comunicații seriale al calculatorului.



În terminologia EIA, modemul este un echipament pentru comunicații de date (Data Communications Equipment – DCE), iar calculatorul este asimilat unui echipament terminal de date (Data Terminal Equipment – DTE). Referințele la DCE și DTE vor apărea în discuțiile referitoare la standardul RS-232C.

Un modem poate comunica într-unul din două moduri:

- *Mod semiduplex*, în care transmisia de date poate surveni într-o singură direcție la un moment dat
 - *Mod duplex*, care permite comunicații independente realizate în ambele sensuri simultan
- Majoritatea modemurilor comunică în mod duplex.

Cablaje RS-232C

Standardul RS-232C mai furnizează semnale de control, cum este Request to Send (cerere de emisie) și Clear to Send (gata de emisie) care pot fi folosite pentru a coordona transmisia și recepția datelor între calculator și modem. Expresia *dialog de confirmare (handshaking)* se referă la coordonarea transferului de date.

Semnalele de confirmare, ca și transmisia și recepția datelor, survin prin intermediul unor fire din cablul de legătură între calculator și modem. Standardul RS-232C specifică un conector pe 25 de pini, cu o funcție specifică atribuită fiecărui pin.

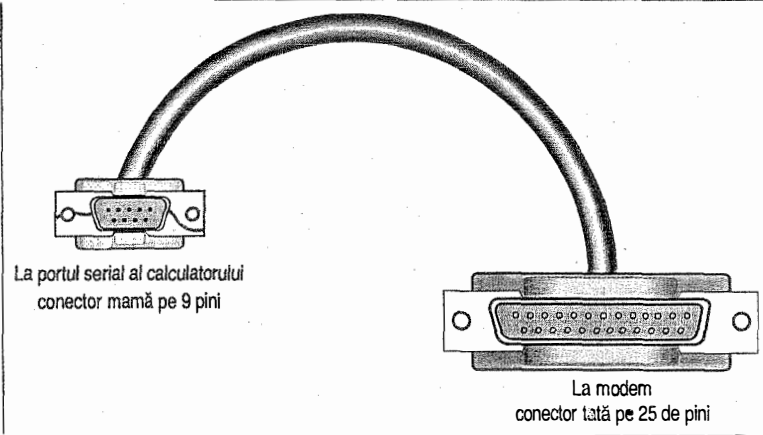


Observație

Un modem obișnuit are un conector mamă trapezoidal pe 25 de pini (denumit conector DB-25), în timp ce portul serial al calculatorului are un conector tată trapezoidal pe 9 pini. Pentru a conecta un port serial al calculatorului la modem, aveți nevoie de un cablu cu un conector mamă DB-9 la un capăt și un conector tată DB-25 la celălalt. Figura 15-2 prezintă un cablu obișnuit de conectare între calculator și modem (a cărui denumire comercială este AT Modem Cable).

Figura 15-2

Un cablu obișnuit pentru conectarea între portul serial al calculatorului și un modem.



Unele calculatoare au un conector DB-25 pentru portul serial. Pentru aceste sisteme, aveți nevoie de un cablu cu conectori DB-25 la ambele capete.

Standardul RS-232C specifică pinii pentru conectorul cu 25 de pini. Figura 15.3 prezintă conectorul DB-25 și funcțiile pinilor pentru conectorii tată pe 25 de pini compatibili IBM PC.

Dintre cei 25 de pini disponibili ai unui conector DB-25, numai 9 sunt folosiți în portul serial al calculatorului. În placa Serial/Parallel Adapter (adaptor serial/paralel) pentru PC-AT, IBM a decis să economisească spațiu prin introducerea unui conector DB-9. Funcțiile pinilor pentru conectorul DB-9 sunt prezentate în figura 15-4.

Figura 15-3

Funcțiile pinilor pentru conectorul serial DB-25.

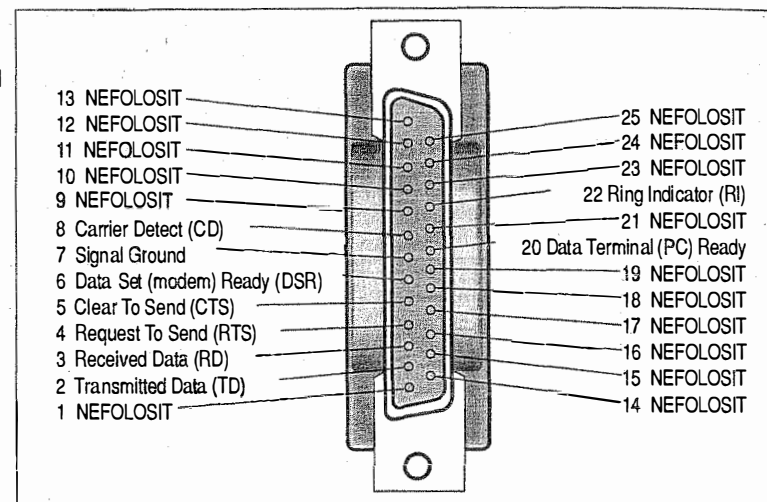
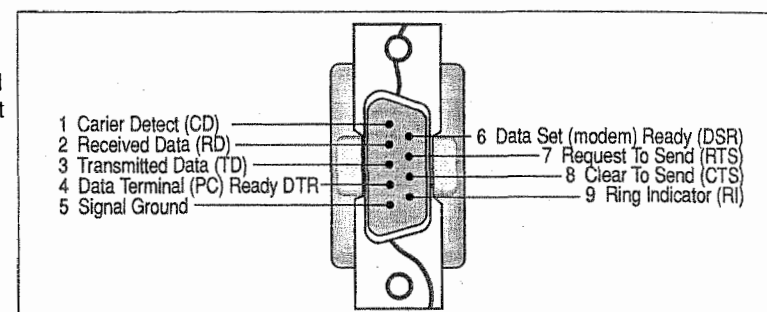


Figura 15-4

Funcțiile pinilor pentru conectorul serial DB-9 folosit în porturile PC seriale.



Utilizarea cablurilor de modem (de la DTE la DCE)

Când conectați un calculator (DTE) la un modem (DCE), cablul trebuie să lege semnalele omonime de la ambele capete după cum urmează (săgeata indică direcția transferului de date pe linia respectivă):

PC (DTE)		Modem (DCE)
TD (Transmit Data)	→	TD (Transmit Data)
RD (Receive Data)	←	RD (Receive Data)
RTS (Request to Send)	→	RTS (Request to Send)
CTS (Clear to Send)	←	CTS (Clear to Send)
DSR (Data Set Ready)	←	DSR (Data Set Ready)
Signal Ground	—	Signal Ground
CD (Carrier Detect)	←	CD (Carrier Detect)
DTR (Data Terminal Ready)	→	DTR (Data Terminal Ready)
RI (Ring Indicator)	←	RI (Ring Indicator)

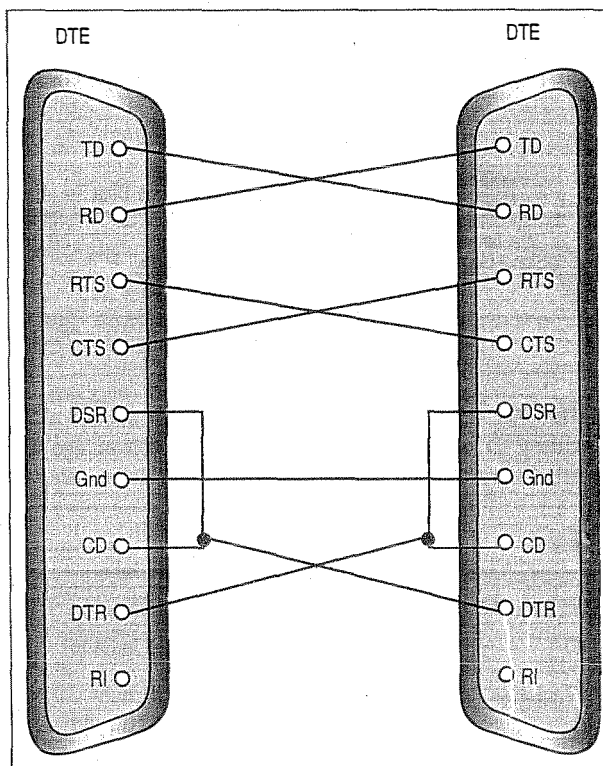
Un cablu care dispune de asemenea conexiuni este frecvent denumit *cablu de legătură directă* (*straight-through cable*), deoarece conectează între ele semnalele similare. Dacă conectorul de la un capăt este DB-25, iar celălalt este un DB-9, tot ce aveți de făcut este să căutați numerele de pini pentru fiecare tip de conector, în figurile 15-5 și 15-6.

Utilizarea cablurilor null-modem

Dacă doriți să conectați două calculatoare prin intermediul unui port serial, nu puteți folosi un cablu de legătură directă, cum este cel folosit pentru a conecta calculatorul la modem. Motivul? Fiecare calculator se așteaptă să trimită date pe linia TD și să primească date pe linia RD. Dacă ambele calculatoare trimit date pe aceeași linie, nici unul nu va fi capabil să-l „audă” pe celălalt. O soluție este crearea unui cablu care să conecteze ieșirea unui calculator (TD) la intrarea celuilalt (RD) și invers. Un asemenea cablu se numește *cablu null-modem*.

Figura 15-5 prezintă interconexiunile semnalelor într-un cablu null-modem obișnuit. În loc de a afișa numerele pinilor, am prezentat numele semnalelor din standardul RS-232C. Numerele exacte ale pinilor depind de tipul conectorilor. Pentru a afla numerele pinilor, examinați figurile 15-3 și 15-4.

Figura 15-5
Interconexiunile
semnalelor într-un
cablu null-modem
obișnuit.



Selecția cablurilor

Când conectați între ele două dispozitive care admit comunicații seriale în conformitate cu standardul RS-232C, selecția cablului depinde de tipul fiecărui dispozitiv: DTE sau DCE.

După cum am arătat anterior, PC-ul este un DTE, iar modemul este un DCE. Multe imprimante și terminale sunt de asemenea DTE-uri.

Pentru a conecta un DTE la un DCE trebuie să folosiți un cablu de legătură directă. Pentru a conecta un DTE la alt DTE, aveți nevoie de un cablu null-modem. Multe magazine de calculatoare vând un adaptor null-modem mai mic. Puteți conecta două dispozitive DTE prin utilizarea unui cablu de modem cu un adaptor null-modem.



Secret

Rețineți că interfața serială a familiei de imprimante HP LaserJet este configurată sub forma unui DTE. Pentru a conecta portul serial al calculatorului la o imprimantă LaserJet, aveți nevoie de un cablu null-modem. Din păcate, interfețele seriale cu imprimantele introduc o complicație suplimentară. Producătorii de imprimante interpretează unele dintre semnalele RS-232C într-o manieră specifică imprimantei, deci este necesar să conectați diferite semnale într-un anumit mod pentru a putea folosi cu succes o imprimantă cu interfață serială.

Controlul fluxului

Standardul RS-232C include semnalele RTS/CTS pentru dialogul de confirmare hardware între perifericele de comunicații (cum sunt PC-ul și modemul). În afară de acest control hardware al fluxului, sunt folosite caractere de control ASCII speciale (Ctrl-Q/Ctrl-S sau XON/XOFF) pentru implementarea software a controlului fluxului. Controlul fluxului este o necesitate, deoarece uneori receptorul este incapabil să țină pasul cu viteza de sosire a datelor și trebuie să poată informa emițătorul să oprească transmisia de date, pentru a putea reveni la normal.

Să presupunem că un receptor are un buffer pentru stocarea caracterelor primite. Pe măsură ce bufferul se umple, receptorul poate trimite caracterul XOFF (Ctrl-S) către emițător, prin care se indică necesitatea opririi transmisiei. Dacă emițătorul înțelege semnificația caracterului XOFF, poate opri transmisia datelor.

Apoi, când receptorul golește bufferul, poate trimite un caracter XON (Ctrl-Q) pentru a indica faptul că transmisia se poate relua. Această schemă de control al fluxului este folosită în multe programe de comunicații, datorită simplității sale. Din păcate, când este folosit protocolul XON/XOFF, caracterele XON și XOFF nu mai pot fi incluse în date, datorită semnificației speciale a acestora.

Standarde de modem

Pe măsură ce serviciile Internet și online au devenit tot mai populare, cererea de modemuri care să ofere rate superioare de transfer de date a crescut. Primele modemuri transferau date la viteze de 300 biți pe secundă (bps), în timp ce modemurile cele mai recente pot transfera la viteze de maximum 56600 bps, ceea ce reprezintă o creștere a performanțelor de peste 100 de ori. Pentru a realiza aceste viteze mari de transfer, modemurile folosesc multe trucuri, inclusiv comprimarea datelor înainte de trimiterea acestora. Așa după cum vă puteți aștepta, două modemuri pot comunica cu succes numai dacă ambele înțeleg semnalul care urmează a fi schimbat între ele. În acest moment intră în scenă protocoalele de modem.



Uniunea Internațională de Telecomunicații (ITU) a ratificat numeroase standarde de modem utilizate în prezent. Aceste standarde au nume care încep cu V. Standardul cel mai recent este V.90, care suportă rate de transfer de date de 56.600 bps (56 Kbps sau 56 K). Standardul V.90 este de asemenea cunoscut sub numele de V.PCM, deoarece folosește o metodă cunoscută sub numele de modulare prin impulsuri (Pulse Coded Modulation – PCM). Tabelul 15-3 prezintă unele dintre standardele de modem folosite la ora actuală.



Sugestie

Dacă doriți să cunoașteți textul exact al Seriei V de recomandări ITU (denumirea acestor standarde de comunicații), puteți achiziționa copii de la situl Web ITU, situat la adresa <http://www.itu.ch/itudoc/itu-t/rec/v/>. Acest sit oferă fișiere în limbile engleză, franceză sau spaniolă și într-un număr de formate de fișier între care amintim Microsoft Word for Windows, Adobe Portable Document Format (PDF) și PostScript.

Tabelul 15-3 Standarde de modem

Nume standard	Rata maximă de transfer de date (bps)
Bell 103	300
Bell 212A	1200
V.17 (Group III Fax)	14400
V.21	300
V.22	1200
V.22bis	2400
V.23	1200
V.27ter (Group III Fax)	4800
V.29 (Group III Fax)	9600
V.32	9600
V.32bis	14400
V.34	28800
V.90 (denumit și V.PCM)	56600

Când achiziționați un modem, verificați dacă modemul se conformează acestor standarde internaționale. În prezent, modemurile de 56 K compatibile ITU sunt la modă.

Comenzi de modem (comenzi AT)

În prezent faimosul set de comenzi AT a apărut pentru prima dată la modemul *Smartmodem* de 300 baud al firmei Hayes, termen creat și brevetat de Hayes Microcomputer Products, Inc. Acest tip de modem funcționa în două moduri distincte:

- **Modul comandă.** În acest mod, caracterele trimise de la calculator (DTE) sunt interpretate ca și comenzi pentru modem.

- **Modul online.** După primirea unei comenzi dial (apelare telefonică) și stabilirea unei conexiuni, modemul trimite toate datele primite în exterior, pe linile telefonice.

Comenzile specifice modemului Hayes Smartmodem încep cu cele două caractere AT (de la *atenție*). Setul inițial de comenzi le includea pe cele de formare a unui număr de telefon, de activare/dezactivare a difuzorului modemului și de setare a modemului pentru a răspunde unui apel.

Sistemul de comenzi AT a fost copiat pe scară largă de producătorii de modemuri, mecanismul de comenzi AT devenind astfel un standard *de facto*. Deși practic toate modemurile folosesc un set fundamental de comenzi, fiecare producător își are propriile sale comenzi patentate, care controlează unele dintre facilitățile exotice sau mai complexe ale modemului.



Sugestie

Am descoperit că este utilă cunoașterea a cel puțin câtorva comenzi AT pentru a controla un modem. Deși multe programe de comunicații ascund detaliile comenzilor AT, puteți ajunge într-o situație în care programul de comunicații este primitiv, reducându-se la trimiterea la celălalt modem a tuturor datelor introduse de la tastatură. În asemenea situații, puteți introduce comenzi AT pentru a configura modemul, pentru a forma numărul de telefon și pentru a stabili legătura. În următoarele secțiuni vom discuta pe scurt despre setul de comenzi AT.

Linia de comandă AT

Așa cum o arată și numele, fiecare comandă din setul de comenzi AT începe cu literele AT. După aceste litere se pot introduce una sau mai multe comenzi valide, după care puteți încheia linia de comandă cu un retur de car (apăsăți Enter pe tastatura calculatorului). Astfel, linia de comandă are următorul format:

```
AT[comanda1][comanda2]...<CR>
```

[comanda1] și [comanda2] reprezintă comenzi opționale, fiecare cu argumente adecvate. Caracterul <CR> din final este returul de car impus.

Să presupunem că doriți să folosiți următoarele comenzi:

- Comanda E cu argumentul 1 forțează modemul să reflecte comenzile de ecou
- Comanda V cu argumentul 1 forțează modemul să furnizeze coduri de rezultat sub formă comunicativă (în locul codurilor numerice)

Puteți trimite aceste comenzi către modem folosind următoarea linie de comandă AT:

```
ATE1V1
```

Așa cum se procedează cu toate liniile de comandă AT, trebuie să încheiați această comandă prin apăsarea tastei Enter. Dacă introduceți această comandă prin intermediul unui program de comunicații, veți vedea că modemul răspunde cu șirul OK.



Secret

Toate modemurile acceptă minim 40 de caractere pe linie de comandă, în care intră și caracterele AT și returul de car din final. Multe modemuri, totuși, pot accepta maximum 255 de caractere pe o linie de comandă AT.

Comanda A/

Fiecare regulă își are excepțiile sale, iar comanda A/ este o excepție de la sintaxa de comandă AT. Dacă introduceți A/ drept singură comandă pe o singură linie (fără a apăsa Enter), modemul repetă imediat ultima comandă primită.

Comenzi de configurare

Aceste comenzi specifică modul de operare al modemului și modul în care acesta răspunde la comenzi. În continuare sunt prezentate câteva comenzi de configurare utile.

- **Comenzile de ecou.** ATE1 determină afișarea comenzii în timpul introducerii acesteia; ATE0 dezactivează afișarea comenzii.

- **Volumul difuzorului.** Comanda ATLn (n este un număr cuprins între 0 și 3) setează volumul difuzorului încorporat al modemului. ATLO și ATL1 setează un volum redus, ATL2 setează un volum mediu, iar ATL3 setează un volum înalt.

- **Controlul difuzorului.** Comanda ATMn (n este un număr cuprins între 0 și 2) controlează dacă și când este activat difuzorul modemului. ATM0 dezactivează difuzorul, ATM1 îl activează până la stabilirea unei legături, iar ATM2 îl activează în permanență.

- **Modul silențios.** La activarea modului silențios (quiet mode), modemul nu recunoaște comenzile și nu raportează starea apelurilor. ATQ0 dezactivează modul silențios și determină modemul să răspundă la comenzi și să afișeze starea apelurilor. ATQ1 activează modul silențios.
- **Modul comunicativ.** La activarea modului comunicativ, modemul recunoaște comenzile și raportează starea apelurilor folosind cuvinte. În caz contrar, modemul răspunde cu coduri numerice (care sunt mai adecvate pentru programele de comunicație decât pentru operatorii umani). Comanda ATV1 activează modul comunicativ, iar comanda ATV0 îl dezactivează. Un modem normal are cele nouă răspunsuri prezentate în tabelul 15.4.

Tabelul 15.4 Răspunsurile unui modem obișnuit

Răspuns numeric	Răspuns în cuvinte
0	OK
1	CONNECT
2	HUNG
3	NO CARRIER
4	ERROR
5	CONNECT 1200
6	NO DIALTONE
7	BUSY
8	NO ANSWER

Majoritatea modemurilor mai folosesc și alte răspunsuri, pentru a raporta conexiuni reușite la rate de transfer superioare.

- **Selecția codului de rezultat.** Comanda ATXn selectează tipul de rapoarte pe care trebuie să le trimită modemul. Argumentul *n* poate lua una dintre următoarele valori:

0	CONNECT
1	CONNECT <i>biti-pe-secunda</i>
2	CONNECT <i>biti-pe-secunda</i> , NO DIALTONE
3	CONNECT <i>biti-pe-secunda</i> , BUSY
4	CONNECT <i>biti-pe-secunda</i> , NO DIALTONE, BUSY

- **Vizualizarea profilurilor memorate.** Utilizarea comenzii AT&V determină modemul să afișeze valorile curente ale unui set selectat de parametri de configurare și valorile regiștrilor interni. Unele modemuri dispun de memorie nonvolatilă pentru a stoca grupuri de setări cunoscute sub numele de *profiluri*. Pe asemenea modemuri, AT&V afișează profilurile memorate.

Comenzi de acțiune

Fiecare comandă de acțiune determină modemul să execute imediat o anumită operație. Cea mai importantă comandă de acțiune este comanda dial (formare a numărului de telefon): ATDT*numar*. Alte două comenzi de acțiune utile sunt ATZ*n* și AT&F*n*, care resetează configurația modemului. În continuare sunt prezentate câteva comenzi de acțiune importante:

- **Apel prin impulsuri.** Comanda ATDP*numar* determină modemul să folosească sistemul de apel prin impulsuri (pulse dial) pentru a forma un anumit număr de telefon. Acest sistem era folosit de telefoanele cu disc. În prezent, pentru apelarea unui număr de telefon se folosește comanda ATDT (apelarea folosind sistemul cu ton) (N.R. – În România se folosesc impulsuri).

- **Apel cu ton.** Comanda ATDT*numar* se folosește pentru apelarea unui număr de telefon folosind sistemul de apel cu ton. De exemplu, pentru a forma numărul 555-1234 se folosește comanda ATDT555-1234. Trebuie să introduceți toate cifrele de care aveți nevoie pentru a forma numărul dorit. Dacă trebuie să formați 9 pentru a accesa o linie exterioară, folosiți comanda ATDT9,555-1234. Virgula introduce o scurtă pauză (de regulă două secunde), necesară pentru a obține o linie cu exteriorul.
- **Formarea ultimului număr.** Comanda ATDL determină modemul să execute ultima comandă de formare a unui număr de telefon.
- **Controlul furcii.** Comanda ATH simulează operația de punere în, respectiv ridicare din furcă a receptorului unui telefon. ATH0 pune receptorul în furcă, iar ATH1 determină modemul să intre pe fir (ca și cum ar fi fost ridicat telefonul din furcă).
- **Răspunsul la apeluri.** Comanda ATA se folosește pentru a determina modemul să răspundă la telefon. Modemul se pune în mod răspuns (prin setarea regiștrului S0 – o zonă de stocare a modemului – la o valoare nenulă) pentru a putea răspunde la eventualele apeluri telefonice. Cu comanda ATA, puteți forța modemul să răspundă la telefon chiar dacă regiștrul S0 este setat la 0 (ceea ce înseamnă că modemul nu va răspunde la telefon).
- **Revenire pe fir.** Comanda ATO readuce modemul la modul online. Folosiți această comandă după ce ați apăsător +++ rapid (introduceți rapid trei semne plus, cu o pauză înainte și după secvență) pentru a scoate modemul din rețea.
- **Resetare software.** Dacă modemul stochează profiluri de configurare în memoria nonvolatilă, puteți reaminti unul dintre profilurile de configurare folosind comanda ATZ*n* (unde *n* este numărul profilului de configurare). Dacă introduceți ATZ fără nici un argument, modemul este resetat. Comanda ATZ determină întreruperea conexiunii existente.
- **Parametri prestabiliți de producător.** Comanda AT&F determină modemul să restabilească setările prestabilite de producător. Unele modemuri folosesc la această comandă un argument numeric; consultați documentația modemului dumneavoastră pentru a primi mai multe informații referitoare la semnificația argumentului numeric.

Comenzile ATSr=*n*

În afară de setul de comenzi AT, modemul Hayes Smartmodem a determinat începutul utilizării regiștrilor interni ai modemului pentru configurarea acestuia. Toate modemurile actuale dispun de un set de registre, denumit *registrele S*, care controlează multe dintre aspectele modemului (inclusiv facilități unice pentru o anumită marcă de modem).



Secret

Un modem obișnuit poate avea între 30 și 60 de registre S, notați prin S0, S1, S2 etc. Comanda ATSr=*n* setează regiștrul S cu numărul *r* la valoarea *n*. Pentru a vizualiza conținutul curent al regiștrului S cu numărul *r*, folosiți comandă ATSr?

Regiștrul S0, de exemplu, conține numărul de sunete de sonerie după care modemul răspunde la telefon. Când S0 este 0, modemul nu răspunde deloc la telefon. Iată cum puteți interoga și seta regiștrul S0 folosind comanda ATS:

```
ATS0?
000
```

```
OK
ATS0=1
OK
ATS0?
001
```



Setul exact de registre S variază de la o marcă de modem la alta, dar majoritatea modemurilor furnizează și interpretează următoarele 13 registre S în aceeași manieră:

- S0 Ring to Answer On** (număr de apeluri după care se răspunde) – Numărul de apeluri după care modemul răspunde la telefon. Când S0 este zero, modemul nu răspunde la telefon.
- S1 Counts Number of Rings** (numărător de apeluri) – Numărul apelurilor primite. Când S1 devine egal cu S0, modemul răspunde la telefon (presupunând că S0 este diferit de zero). Modemul resetează S1 la zero după ultimul apel.
- S2 Escape Code Character** (caracter cod escape) – Caracterul folosit drept secvență escape pentru a comuta modemul din mod online în mod comandă. Valoarea prestabilită este 43, care este codul ASCII pentru caracterul plus (+). Pentru a comuta din modul online în mod comandă, trebuie să introduceți acest caracter de trei ori, într-o succesiune rapidă.
- S3 Carriage-Return Character** (caracter retur de car) – Codul ASCII al caracterului folosit drept retur de car (cu acest caracter se încheie linia de comandă AT). Valoarea prestabilită este 13.
- S4 Line-Feed Character** (caracter salt la linie nouă) – Codul ASCII al caracterului folosit drept caracter salt la linie nouă atunci când modemul generează răspunsuri cu cuvinte. Valoarea prestabilită este 10.
- S5 Backspace Charater** (caracter backspace) – Codul ASCII al caracterului folosit drept caracter backspace. Modemul reflectă acest caracter pentru a implementa funcția de ștergere a caracterului anterior. Valoarea prestabilită este 8.
- S6 Wait Time for Dial Tone** (timp de așteptare a tonului de apel) – Numărul de secunde așteptate înainte de formarea primei cifre dintr-un număr de telefon. Valoarea prestabilită este 2.
- S7 Wait Time for Carrier** (timp de așteptare a purtătoarei) – Numărul de secunde cât modemul așteaptă purtătoarea. Dacă modemul nu detectează o purtătoare după acest interval de timp, emite mesajul NO CARRIER. Valoarea prestabilită depinde de modem, dar variază undeva între 30 și 60.
- S8 Comma Time** (timp de virgulă) – Intervalul (în secunde) de pauză când modemul găsește o virgulă în numărul de telefon de format. Valoarea prestabilită este 2.
- S9 Carrier Detect Time** (interval de detecție al purtătoarei) – Intervalul de timp (în zecimi de secundă) în care purtătoarea trebuie să fie prezentă înainte ca modemul să declare că a fost detectată o purtătoare. Valoarea prestabilită este 6, ceea ce înseamnă că purtătoarea trebuie să fie prezentă timp de 0,6 secunde înainte de a fi detectată de modem.
- S10 Carrier Loss Time** (interval de pierdere al purtătoarei) – Intervalul de timp (în zecimi de secundă) în care se poate pierde purtătoarea fără ca modemul să se deconecteze. Valoarea prestabilită este cuprinsă între 7 și 15, ceea ce înseamnă că purtătoarea poate fi pierdută între 0,7-1,5 secunde, fără a provoca deconectarea modemului.
- S11 Dial-Tone Spacing** (intervalul dintre tonuri) – Durata (în milisecunde) a fiecărui ton de apel și intervalul de timp între tonurile adiacente. Valoarea prestabilită este cuprinsă undeva între 50 și 100 milisecunde (50 se consideră ca fiind minimul necesar pentru recunoașterea tonurilor de apel de către sistemul telefonic).

S12 Escape Sequence Guard Time (timpul de atenție) – Intervalul de timp (în 1/50 dintr-o secundă) care trebuie să survină înainte și după secvența escape (secvența prestabilită este ++++) care comută modemul între modul online și modul comandă. Valoarea prestabilită este 50, ceea ce înseamnă că timpul de atenție este de o secundă.

Asistență online

Ca răspuns la comanda AT\$, modemurile U.S. Robotics afișează informații de asistență online la seturile elementare de comenzi. Informația de asistență este instructivă, deoarece afișează gama de comenzi acceptate de modem. Comanda respectivă poate fi introdusă într-un program de comunicații seriale cum este Minicom, pe care îl voi descrie pe scurt în secțiunea „Apelarea în exterior cu un program de comunicații”

Linux și modemurile

Dacă folosiți Linux la domiciliu sau într-o firmă de mici dimensiuni, modemul poate fi folosit într-unul din următoarele scopuri:

- Pentru a stabili legătura cu alt calculator (cum ar fi un sistem de buletine electronice), cu un serviciu online (cum este CompuServe) sau un alt sistem UNIX, din firma sau universitatea unde lucrați.
- Pentru a permite altor persoane să vă apeleze sistemul și să vă folosească sistemul Linux. Dacă calculatorul dumneavoastră personal de la domiciliu rulează Linux și aveți un modem configurat, vă puteți apela propriul sistem de la birou.
- Pentru a folosi intrarea în rețea prin apel telefonic cu Serial Internet Protocol (SLIP) sau Point-to-Point Protocol (PPP) în vederea conectării la Internet (de regulă prin intermediul unui furnizor de servicii Internet).



Referință încrucișată

În secțiunile următoare ne vom ocupa de primele două utilizări ale unui modem: apelarea sistemului dumneavoastră Linux PC (dial-in) sau apelarea de către acesta a altor sisteme sau servicii (dial-out). Conectarea în rețea prin apel telefonic folosind SLIP sau PPP este un subiect important în sine; vom discuta în detaliu despre acesta în capitolul 18.

Apelarea altor sisteme sau servicii prin intermediul unui modem

Când ați instalat Linux de pe CD-ROM-ul anexat acestei cărți, ați instalat în mod automat unele instrumente pe care le puteți folosi pentru a apela sisteme sau servicii din sistemul dumneavoastră Linux, prin intermediul unui modem. Înainte de a realiza aceste apeluri, totuși, trebuie să verificați dacă dispuneți de un modem conectat în mod adecvat la unul dintre porturile seriale ale calculatorului dumneavoastră și că perifericele Linux pentru porturile respective sunt corect setate.

Examinarea configurării hardware a unui modem

Verificați dacă modemul este conectat în mod corespunzător la sursa de alimentare cu energie electrică și că este de asemenea conectat la linia telefonică.

Cumpărați cablul adecvat pentru conectarea modemului la calculator. Așa cum s-a explicat în secțiunile anterioare ale acestui capitol, aveți nevoie de un cablu serial cu legătură directă pentru a conecta modemul la calculator. Conectorii de la extremitățile cablului depind de tipul

de conector serial al calculatorului dumneavoastră. Extremitatea pentru modem a cablului necesită un conector tată DB-25. Extremitatea pentru calculator este frecvent un conector mamă DB-9, dar unele calculatoare mai necesită un conector mamă DB-25 la extremitatea pentru calculator a cablului.

Cablurile de modem se găsesc în majoritatea magazinelor de calculatoare. În particular, cablurile de modem DB-9 și DB-25 se vând deseori sub denumirea „AT Modem Cable”.



Atenție!

Dacă portul serial al calculatorului dumneavoastră este un DB-25, conectorul din partea din spate a calculatorului (nu cel de pe cablu) este un conector tată DB-25. Nu confundați acest conector cu conectorul DB-25 al portului paralel, care este conector mamă. Dacă folosiți greșit conectorii, nu se va produce nici o pagubă, dar comunicația serială nu va avea loc.



Sugestie

Dacă calculatorul dumneavoastră personal are modem intern, tot ce aveți de făcut este să vă asigurați că IRQ-ul și adresele I/O sunt corect setate (presupunând că placa de modem are jumpere pentru setarea acestor valori). Pentru COM1, setați IRQ la 4 și adresa I/O la 0x3f8; pentru COM2, IRQ este 3 și adresa I/O este 0x2f8. De asemenea, trebuie să conectați linia telefonică la o mufă de telefon plasată în spatele plăcii de modem intern.

Examinarea perifericelor seriale Linux

Când instalați Linux de pe CD-ROM-ul anexat acestei cărți, respectând indicațiile din capitolul 1, perifericele seriale Linux necesare sunt create în mod automat. Trebuie să dispuneți de două seturi de periferice: /dev/cua/* (* este numărul portului) pentru apelarea în exterior prin modem și /dev/ttyS* pentru apelul din exterior al sistemului dumneavoastră tot prin intermediul modemului. Perifericele /dev/ttyS* se pot folosi pentru apelurile în ambele sensuri.



Văjitor

Procesul de instalare creează fișierele /dev/cua* și /dev/ttyS* cu o setare de permisiune care nu permite oricărui utilizator să citească perifericul. Dacă doriți să permiteți tuturor utilizatorilor să poată apela în exterior prin modem, tastați următoarea comandă după intrarea în sistem ca root:

```
chmod o+rw /dev/ttyS*
```

Această comandă oferă tuturor utilizatorilor accesul la perifericele pentru apeluri spre exterior.

Apelarea în exterior folosind un program de comunicații

După ce ați încheiat instalarea fizică a modemului și verificați existența fișierelor Linux de periferic necesare, puteți încerca să apelați în exterior folosind modemul. Cea mai bună metodă este utilizarea programului de comunicații seriale minicom, inclus în distribuția Linux existentă pe CD-ROM-ul anexat acestei cărți. Programul minicom, creat de Miquel van Smoorenburg, este un program de comunicații cu o interfață bazată pe text care emulează un terminal VT102.

Pentru a rula minicom, tastați **minicom** la promptul interpretorului într-o fereastră xterm sau într-un ecran Virtual Terminal. Dacă rulați minicom ca utilizator normal (nu root), minicom poate afișa următorul mesaj de eroare și își va încheia execuția:

```
minicom
minicom: there is no global configuration file /etc/minirc.dfl
Ask your sysadm to create one (with minicom -s).
```

(Nu există nici un fișier de configurare globală /etc/minirc.dfl. Solicitați administratorului de sistem crearea fișierului, folosind comanda minicom -s.)

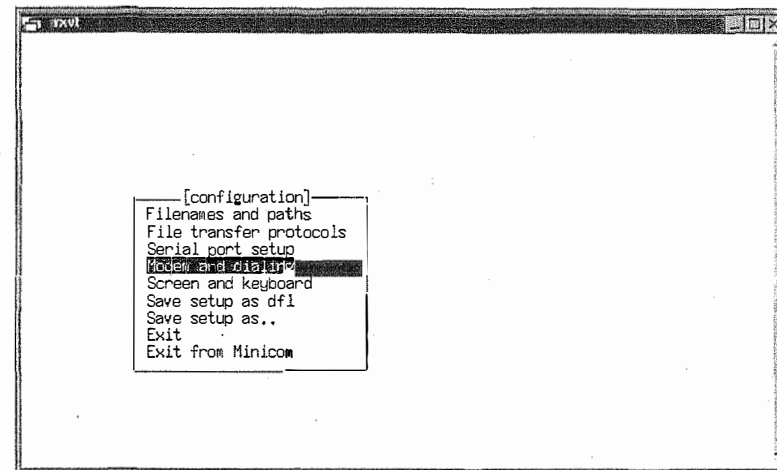
Intrați în sistem ca root și apoi tastați

```
minicom -s
```

Minicom se lansează și afișează o casetă de dialog care vă permite să configurați diferite aspecte ale programului, inclusiv portul serial și comenzile de formare a numărului de telefon prin modem. Pentru a introduce comenzile de inițializare ale modemului, folosiți tasta cu săgeată pentru a evidenția opțiunea Modem and dialing, așa cum se vede în figura 15-6.

Figura 15-6

Meniul de configurare al programului Minicom.



Având opțiunea Modem and dialing evidențiată, apăsați Enter. Minicom va afișa un alt meniu, cu o listă de setări care pot fi modificate. Apăsați A pentru a modifica șirul de inițializare al modemului. Minicom așază cursorul pe prima linie, așa cum se vede în figura 15-7.

Apoi, puteți folosi tasta Backspace pentru a edita linia respectivă și pentru a introduce comanda AT necesară pentru inițializarea modemului. De exemplu, eu dispun de un modem de 56 K, care folosește standardul învechit (în prezent) x2. Pentru a mă conecta la modemul furnizorului meu de servicii Internet, trebuie să trimit următoarea comandă AT:

```
AT &F1 &K0 &IO s33=32 s15=28 s27=64 s32=64
```

După ce ați introdus șirul de inițializare a modemului, apăsați Enter pentru a reveni la meniul de nivel maxim. Dacă nu mai aveți modificări de operat, folosiți tastele cu săgeți sus și jos pentru a evidenția opțiunea Save as dfl (salvează în format dfl) – ceea ce înseamnă salvare ca opțiune prestabilită – și apoi apăsați Enter. Minicom salvează setările în fișierul /etc/minirc.dfl, care este fișierul de configurare prestabilit pentru Minicom. După aceasta, puteți părăsi Minicom prin selectarea opțiunii Exit from Minicom, urmată de apăsarea tastei Enter.

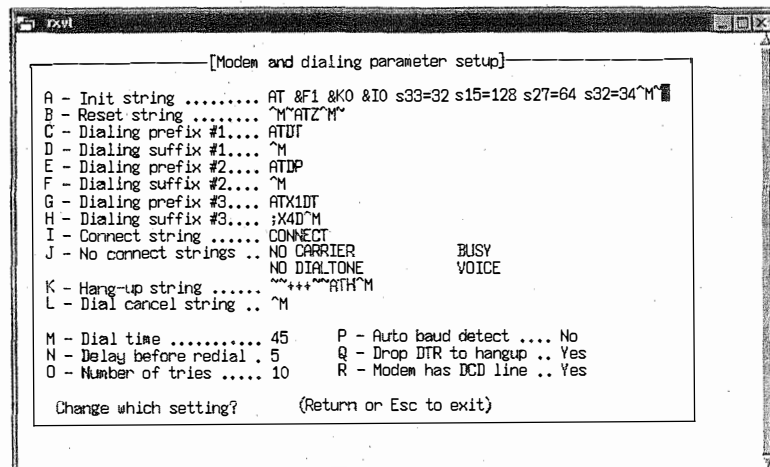
De asemenea, mai aveți două lucruri de făcut înainte de a permite oricărui utilizator să ruleze minicom:

- Editați fișierul text /etc/minicom/users și adăugați o linie nouă cu cuvântul ALL pentru a permite tuturor utilizatorilor să acceseze fișierul de configurare minicom.

- Permiteți tuturor utilizatorilor să citească din și să scrie în portul serial unde este conectat modemul. Dacă modemul se află pe /dev/ttyS0, trebuie să introduceți următoarea comandă:

```
ln -s /dev/ttyS0 /dev/modem
```

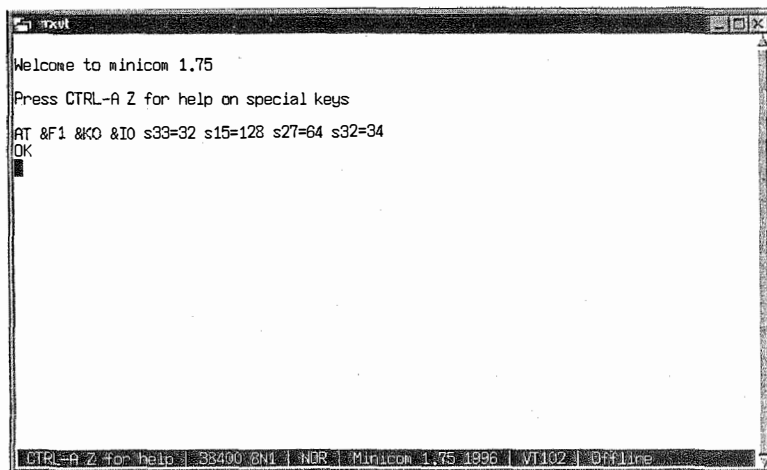
Figura 15-7
Configurarea
parametrilor de
modem și de
apelare în
Minicom



După aceasta, puteți rula minicom ca utilizator obișnuit. La prima rulare a programului minicom, acesta va reseta modemul. Figura 15-8 prezintă rezultatul rulării programului minicom într-o fereastră xterm.

Programul minicom este oarecum similar cu alt program de comunicații shareware, denumit Telix. Similar cu programul respectiv, puteți apăsa Ctrl-A pentru a obține atenția programului minicom. După ce apăsați Ctrl-A, dacă apăsați Z, va apărea un ecran de asistență sub forma unei ferestre text, așa cum se vede în figura 15-9.

Figura 15-8
Ecranul inițial al
programului
Minicom.



Secret

Din ecranul de asistență puteți obține informații despre alte comenzi ale programului minicom. Din ecranul de asistență, apăsați Enter pentru a reveni la modul online. În modul online, puteți folosi comenzile AT ale modemului pentru a executa apeluri în exterior. În particular, puteți folosi comanda ATDT pentru a forma numărul de telefon al unui alt modem (de exemplu, numărul furnizorului

dumneavoastră de servicii Internet sau al sistemului de la birou), așa cum se vede în figura 15-10. O dată ce ați ajuns la promptul de login, puteți intra în sistem ca de obicei și puteți folosi sistemul aflat la distanță.

Figura 15-9
Ecranul de
asistență al
programului
Minicom.

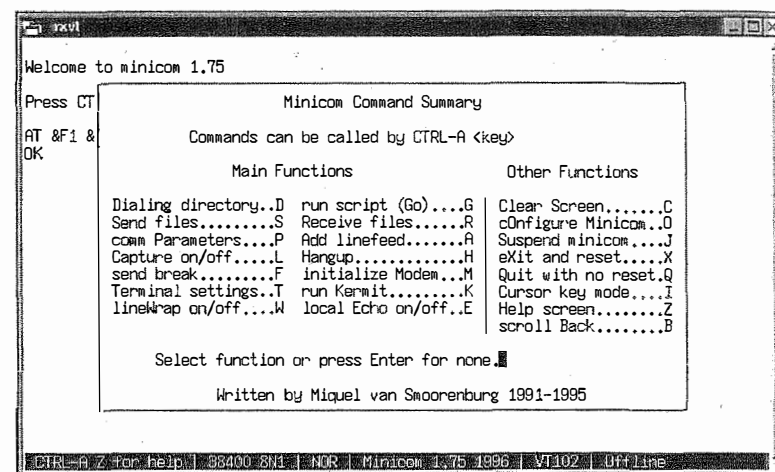
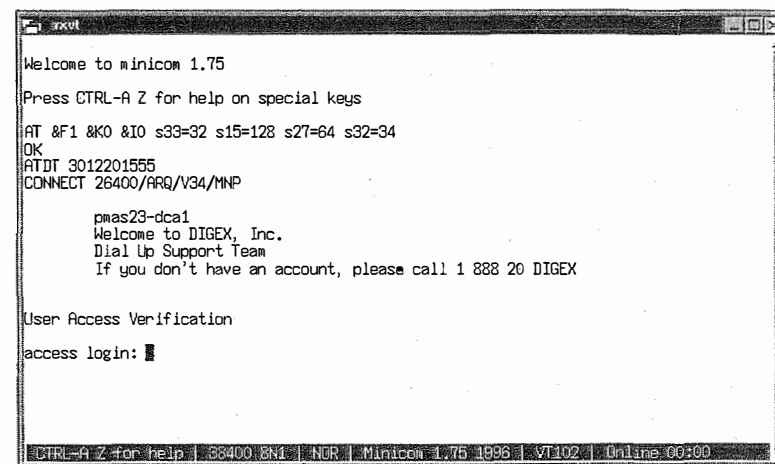


Figura 15-10
Apelarea unui
sistem aflat la
distanță folosind
programul
Minicom



Când ieșiți din celălalt sistem și doriți să părăsiți programul minicom, apăsați Ctrl-A și apoi tastați X pentru a părăsi programul. Apăsați din nou Enter pentru a răspunde la promptul minicom, așa cum se vede în figura 15-11.

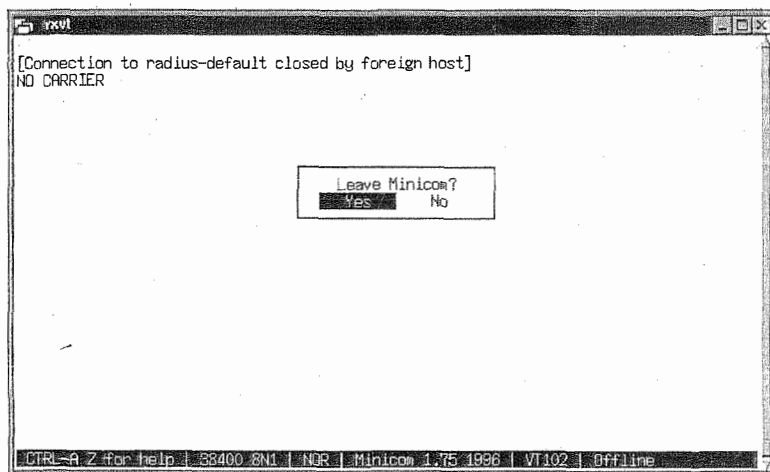
Configurarea Linux pentru apelarea din exterior (dial-in)

Același modem poate fi configurat pentru apeluri din exterior, ca și pentru apeluri spre exterior (desigur, numai una dintre aceste operații este permisă la un moment dat). Etapele de configurare a unui modem pentru primirea apelurilor implică configurarea unui program care

să monitorizeze portul serial în așteptarea unor apeluri din exterior; acest program furnizează promptul de login. Numele generic al acestor programe este `getty`. Puteți folosi programul `ugetty` inclus în distribuția Linux de pe CD-ROM-ul anexat acestei cărți.

Figura 15-11

Acest prompt `minicom` apare când apăsați `Ctrl-A` și apoi `X` pentru a părăsi programul.



În următoarele câteva secțiuni este explicat modul de configurare a unui modem pe COM1 pentru recepția de apeluri. Numele de periferic Linux pentru modemul respectiv (când este configurat pentru recepția apelurilor) este `/dev/ttyS0`.

Editarea fișierului `/etc/rc.d/rc.serial`

La lansarea Linux, procesul `init` execută scriptul `/etc/rc.d/rc.sysinit`, care la rândul său execută scriptul `/etc/rc.d/rc.serial`, dacă acesta există. Astfel, puteți configura porturile seriale prin inserția comenzilor adecvate în fișierul `/etc/rc.d/rc.serial` (pur și simplu creați fișierul `/etc/rc.d/rc.serial`, dacă acesta nu există).

Dacă dispuneți de un modem de viteză mare, de exemplu un modem cu 56 K, trebuie să inserați o comandă în fișierul `rc.serial` pentru a rula programul `setserial` și a seta indicatorul `spd_vhi` pentru portul serial. Setarea indicatorului `spd_vhi` permite calculatorului dumneavoastră să comunice cu modemul la o viteză cu mult mai mare decât cea folosită de modem pentru a se conecta cu un sistem aflat la distanță. Inserați următoarea linie în fișierul `rc.serial`:

```
/bin/setserial /dev/ttyS0 spd_vhi
```



Secret

Dacă modemul dumneavoastră suportă componente hardware care folosește semnalele RTS (Request to Send) și CTS (Clear to Send), atunci trebuie să le folosiți. Activați dialogul de confirmare hardware pentru modem cu comanda `stty` prin inserția următoarei comenzi în fișierul `rc.serial`:

```
/bin/stty crtscts < dev/ttyS0
```

Actualizarea fișierului de configurare al programului `ugetty`

Programul `ugetty` verifică fișierul `/etc/gettydefs` în căutarea unor informații asupra vitezei și a setărilor unui port serial (frecvent denumit linie). Intrările din fișierul `/etc/gettydefs` indică de asemenea programului `ugetty` tipul de prompt de login afișat.



Fișierul `/dev/gettydefs` ar trebui să funcționeze fără modificări. Cu toate acestea, se poate edita promptul de login. Eu am folosit următoarea setare pentru un modem:

```
# 38400 fixed-baud modem entry
F38400# B38400 CS8 CRTSCS # B38400 SANE -ISTRIP HUPCL CRTSCS #es
Login: #F38400
```

Fișierul `/etc/gettydefs` are cu mult mai multe setări, dar eu am folosit-o numai pe aceasta. Eticheta `F38400` de la începutul liniei este ceea ce se specifică la rularea programului `ugetty` pe linia `/dev/ttyS0`.

Indicatorul `CRTSCS` arată faptul că modemul folosește un control hardware al fluxului, pe baza semnalelor RTS/CTS (pe care le-am activat cu comanda `stty` în fișierul `/etc/rc.d/rc.serial`).

Această intrare se afla deja în fișierul `/etc/gettydefs`. Articolele din apropierea ultimei linii sunt destinate promptului de login. Puteți folosi codurile prezentate în tabelul 15-5 pentru a afișa anumite informații în promptul de login.

Tabelul 15-5 Unele coduri utile din fișierul `/etc/gettydefs`

Cod	Semnificație
@B	Rata curentă de transfer de date, în biți pe secundă.
@D	Data curentă, în format LL/ZZ/AA.
@L	Linia serială la care este atașat <code>ugetty</code> (cum ar fi <code>ttyS0</code>)
@S	Numele sistemului (așa cum apare la comanda <code>uname -n</code>)
@T	Ora curentă, în format OO:MM:SS (format 24 de ore)
@U	Numărul curent de utilizatori în sistem
@V	Văloarea variabilei <code>VERSION</code> , specificată în fișierul de configurare (<code>/etc/conf.ugetty.ttyS0</code> pentru <code>ugetty</code> care rulează pe linia <code>ttyS0</code>).

Pentru a afișa un singur caracter `@` la promptul login, folosiți `\@` sau `@@`. Puteți insera și alte caractere la promptul de login prin utilizarea escape-ului cu backslash. Tabelul 15-6 prezintă câteva dintre caracterele escape recunoscute. Cu aceste facilități, puteți personaliza promptul de login în conformitate cu necesitățile dumneavoastră

Tabelul 15-6 Caractere escape în fișierul `/etc/gettydefs`

Caracter	Semnificație
\\	Backslash simplu
\b	Cracater backspace (Ctrl-H)
\c	Oprește linia nouă la finalul unui șir
\f	Salt de pagină (Ctrl-L)
\n	Caracter linie nouă (Ctrl-J)
\s	Spațiu

(continuare)

Tabelul 15.6 Continuare

Caracter	Semnificație
\t	Spațiu de tabulare
\nnn	Caracterul ASCII a cărui valoare zecimală este <i>nnn</i> (dacă numărul începe cu 0, se presupune că valoarea este în octal; dacă începe cu 0x, este considerată ca fiind în hexazecimal)

Pregătirea fișierului de configurare a programului uugetty

Programul uugetty are numeroși parametri care pot fi modificați. De fapt, puteți specifica opțiuni separate pentru uugetty pe bază de linii. Aceste fișiere de configurare per linie se găsesc în directorul `/etc`. Numele fiecărui fișier de configurare este de forma `uugetty.linie`, unde *linie* este numele de periferic al liniei (fără partea cu `/dev`). Astfel, `/etc/conf.uugetty.ttyS0` este fișierul de configurare pentru copia programului uugetty care rulează pe linia `/dev/ttyS0`.

Iată conținutul unui fișier `/etc/default/uugetty.ttyS0` obișnuit de pe unul din sistemele Linux PC ale autorului:

```
# Line to initialize (line name should not include /dev)
INITLINE=cua0
# Alternate lock file to check (if this lock file exists,
# then uugetty is restarted so that the modem is re-
# initialized)
ALTLOCK=cua0

# Timeout to disconnect if idle...
TIMEOUT=60

# Modem initialization string. Sets the modem not to auto-
# answer. When a call comes, uugetty will issue an ATA command
# to make the modem go online
# Format of these entries:
# <expect> <send>... (chat sequence)
INIT="" \d+++ \dAT\r OK\r\n ATH0\r OK\r\n
AT\sM0\sE1\sQ0\sV1\sX4\sS0=0\sC1\sS0\r OK\r\n

# Waitfor string. If this sequence of characters is received
# over the line, a call is detected.
WAITFOR=RING

# The following line is the connect chat sequence. This chat
# sequence is performed after the WAITFOR string is found. The
# \A character automatically sets the baud rate to the
# characters that are found, so if you get the message CONNECT
# 26400, the baud rate is set to 26400 baud.
#
# format: <expect> <send> ... (chat sequence)
CONNECT=""ATA\r CONNECT\s\A

# The next line sets the time to delay before sending the login
# banner
DELAY=1
```

Intrările din acest fișier controlează modul în care programul uugetty inițializează modemul, așteaptă un apel și răspunde cu un prompt de login.

Procesul uugetty folosește șirul INIT pentru a inițializa modemul. În acest caz, secvența de inițializare emite următoarele comenzi de modem:

Scopul comenzii

ATMO	Închide difuzorul
ATE1	Activează ecoul comenzilor
ATQ0	Dezactivează modul silențios (pentru ca modemul să raporteze mesajele de stare)
ATV1	Activează modul comunicativ
ATX4	Determină modemul să raporteze coduri de rezultat detaliate
ATS0=0	Dezactivează răspunsul automat (uugetty pune modemul online la momentul potrivit cu o comandă ATA).
AT&C1	Permite modemului să detecteze linia de detecție a purtătoarei (carrier-detect - CD), linie care este activată numai după conectare
AT&S0	Activează semnalul DSR (Data Set Ready) și îl lasă activ

Procesul uugetty așteaptă șirul WAITFOR înainte de a folosi șirul CONNECT pentru a iniția conexiunea. Când modemul primește un apel, afișează mesajul de stare RING (sună). Când uugetty primește șirul RING, trimite modemului șirul CONNECT. Șirul CONNECT conține comanda ATA, care pune modemul online.

După aceea, uugetty afișează conținutul fișierului `/etc/issue`, urmat de promptul de login. Apoi uugetty rulează programul `/bin/login` pentru a se ocupa de login-ul curent.

Lansarea programului uugetty în /etc/inittab

Pentru a vă asigura că uugetty rulează la lansarea Linux, trebuie să inserați o linie în fișierul `/etc/inittab`, pentru ca procesul `/init` să lanseze uugetty în mod automat. În continuare, iată ce am inserat în fișierul `/etc/inittab` pentru a lansa o copie a programului uugetty pe linia `/dev/ttyS0` (pentru modemul de la COM1):

```
# Serial line with dial-in modem
s0:235:respawn:/sbin/uugetty ttyS0 F38400 vt100
```



Referință
Încrucișată

Această comandă se asigură că uugetty rulează pe nivelele 2, 3 și 5. (Consultați capitolul 5, care conține o discuție despre nivelele de rulare și fișierul `/etc/inittab`.)

Argumentele comenzii uugetty au următoarea semnificație:

- `ttyS0` specifică linia monitorizată de uugetty (`ttyS0` înseamnă `/dev/ttyS0`).
- `F38400` specifică viteza liniei (`F38400` trebuie să apară în fișierul `/etc/gettydefs`).
- `vt100` specifică tipul de terminal (acesta trebuie să se regăsească în fișierul `/etc/termcap`).

Testarea configurării pentru primirea apelurilor

După ce ați parcurs toate etapele de configurare pentru uugetty și pentru modem, trebuie să vă asigurați că procesul `init` reexaminează fișierul `/etc/inittab` și lansează uugetty conform specificațiilor din fișierul respectiv. Pentru aceasta, intrați în sistem ca `root` și introduceți următoarea comandă:

```
init q
```

După această comandă, puteți testa modemul la primirea apelurilor. Puteți cere unui prieten să vă apeleze sistemul, pentru a vedea dacă login funcționează. Dacă sunteți atât de norocos încât să dispuneți de două linii telefonice (și mai multe calculatoare la domiciliu), puteți apela în exterior pe o linie și să intrați în sistemul Linux configurat pe cealaltă linie. Pe sistemul meu, iată ce afișează promptul de login pe un PC când apelez sistemul meu Linux:

```
Red Hat Linux release 5.0 (Hurricane)
Kernel 2.0.31 on an i586
```

lnbsoft login:

Fișierul /etc/gettydefs din sistemul meu specifică promptul de login după cum urmează:

```
@S login:
```

Conform acestei linii, promptul trebuie să includă numele sistemului (@S), urmat de șirul login:, adică ceea ce apare pe ecran la apelarea sistemului meu.

Acum, sistemul dumneavoastră Linux este pregătit să permită utilizatorilor să execute login prin intermediul modemului configurat pentru recepția apelurilor.

Terminale și plăci seriale multiport

În secțiunile anterioare a fost prezentat în detaliu modul de configurare și utilizare a unui modem în Linux pentru apelarea altor sisteme, dar și pentru a permite altor persoane să apeleze sistemul. În continuare, vom descrie etapele necesare pentru configurarea terminalelor conectate la portul serial al calculatorului. De asemenea, capitolul de față mai conține o listă cu plăci seriale multiport suportate de Linux.



Observație

Puteți folosi alte calculatoare personale MS-DOS (mai ales sisteme mai vechi 286 și 386) ca terminale conectate la sistemul dumneavoastră Linux. Tot ce aveți nevoie pe aceste PC-uri este un program de comunicații seriale, precum Procomm Plus sau Telix.

Configurarea unui terminal la un port serial

Pentru a configura un terminal pe un port serial, trebuie să setați un proces getty, așa cum ați procedat la configurarea unui modem pentru recepția apelurilor. Parcurgeți următoarele etape:

1. Verificați dacă ați folosit cablul serial corect pentru a conecta terminalul la portul serial. Majoritatea terminalelor necesită un cablu de null-modem în care liniile TD (transmitere date) și RD (recepție date) sunt conectate invers. (Pentru detalii privind cablurile seriale, consultați paragrafele anterioare ale acestui capitol.)

2. Configurați parametri de comunicație ai terminalului. Etapele exacte depind de cipul de terminal.

3. Editați /etc/gettydefs și adăugați o linie la terminal. Dacă terminalul operează la 9600 bps, specificați o linie de genul următor (linia care începe cu # este un comentariu):

```
# 9600 bps connection to a terminal
T9600# B9600 CS8 CLOCAL # 89600 SANE -ISTRIP CLOCAL #@S login:
#T9600
```

Partea cea mai importantă a comenzii este indicatorul CLOCAL, care indică procesului getty să ignore semnalele de control al modemului.

4. Editați fișierul /etc/inittab și adăugați o linie pentru a lansa un proces getty pe linia conectată la terminal. Dacă dispuneți de un terminal compatibil VT-100 pe linia ttys1 (COM2), puteți adăuga următoarea linie la fișierul /etc/inittab din sistemul dumneavoastră:

```
s2:235:respawn:/sbin/getty/ ttys1 T9600 vt100
```

5. Intrați în sistem ca root și forțați procesul init să reexamineze fișierul /etc/inittab, după cum urmează:

```
init q
```

6. Pe ecranul terminalului va apărea un prompt de login; în acest moment puteți intra în sistem.

Configurarea plăcilor seriale multiport în Linux

Dacă intenționați să susțineți o mică afacere cu un sistem Linux PC și terminale simple (terminalele sunt mai ieftine decât calculatoarele complet echipate, deși ca terminale puteți folosi PC-uri mai vechi), atunci aveți nevoie de mai multe porturi seriale. Cu o altă placă serială, calculatorul poate suporta patru porturi seriale. Dacă aveți nevoie de mai mult de patru porturi seriale, trebuie să vă procurați porturi seriale I/O speciale, cunoscute sub numele de *plăci seriale multiport*. Aceste plăci includ suport pentru 4-32 de plăci seriale. Porturile seriale au un singur IRQ, dar fiecare port are o adresă I/O unică.



Sugestie

Multe plăci seriale multiport folosesc cipurile UART 16450 sau 16550A. Când vă procurați o placă, trebuie să vă asigurați că cipul UART este compatibil 16550A.

Pentru a adăuga suport pentru o placă serială multiport normală, va trebui să eliminați specificatorul de comentariu al liniilor corespunzătoare din fișierul /etc/rc.d/rc.serial. Linux suportă următoarele plăci seriale multiport bazate pe cipurile UART 16450 sau 16550A:

- AST FourPort și clonele (4 porturi)
- Accent Async-4 (4 porturi)
- Arnet Multiport-8 (8 porturi)
- Bell Technologies HUB6 (6 porturi)
- Boca BB-1004 (4 porturi), BB-1008 (8 porturi), BB-2016 (16 porturi)
- Boca IOAT66 (6 porturi)
- Boca 2by4 (4 porturi seriale și 2 porturi paralele)
- Computone ValuePort V4-ISA (compatibilă AST FourPort)
- Digi PC/8 (8 porturi)
- GTEK BBS-550 (8 porturi)
- Longshine LCS-8880, Longshine LCS-8880+ (compatibilă AST FourPort)
- Moxa C104, Moxa C104+ (compatibilă AST FourPort)
- PC-COMM (4 porturi)
- Sealevel Systems COMM-2 (2 porturi), COMM-4 (4 porturi) și COMM-8 (8 porturi)
- SIIG I/O Expander 2S IO1812 (4 porturi)
- STB-4COM (4 porturi)
- Twincom ACI/550
- Usenet Serial Board II (4 porturi)

Rețineți că plăcile Boca BB-1004 și BB-1008 nu includ suport pentru semnalele Carrier Detect (CD) și Ring Indicator (RI) necesare pentru funcționarea modemurilor în regim de

primire a apelurilor. Ca atare, nu puteți folosi plăcile BB-1004 și BB-1008 cu modemuri pentru recepția apelurilor.

Unele plăci seriale multiport folosesc procesoare speciale în locul cipurilor UART 16450 și 16550A. Aceste plăci seriale multiport inteligente necesită drivere speciale. Tabelul 15-7 prezintă câteva plăci seriale multiport și locațiile driverelor pentru Linux ale acestora.

Tabelul 15-7 Locațiile driverelor Linux pentru plăcile seriale multiport

Nume	Porturi	Locația driverului
Control RocketPort	4, 8, 16, 32	ftp://tsx-11.mit.edu/pub/linux/packages/control
Cyclades Cyclom-Y	8-32	Inclus în nucleul Linux (activați CONFIG_CYCLADES la configurarea nucleului)
Decision PCCOM8	8	ftp://ftp.signum.se/pub/pccom8
DigiBoard PC/Xx	2-64	Inclus în nucleul Linux (activați CONFIG_DIGI la configurarea nucleului)
Equinox SuperSerial	2-128	ftp://ftp.equinox.com/library/sst
GTEK Cyclone	6, 16, 32	ftp://ftp.gtek.com/pub
Hayes ESP	8	http://www.nyx.net/~arobinso
Maxpeed SS	4, 8, 16	ftp://maxpeed.com/pub/ss
Moxa C218, C320	8-32	ftp://ftp.moxa.com.tw/drivers/c218-320/linux
SDL RISCom/8	8	Inclus în nucleul Linux (activați CONFIG_RISCOM8 la configurarea nucleului)
Specialix SIO	4-32	ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial
Stallion	4-64	Inclus în nucleul Linux (activați CONFIG_STALDRV la configurarea nucleului)



Sugestie

Pentru a învăța mai multe despre configurarea unei plăci seriale multiport, puteți citi cel mai recent document Serial-HOWTO. Pentru a citi documentele HOWTO, tastați `cd /usr/doc/HOWTO` pentru a comuta în directorul unde se află localizate fișierele HOWTO. Apoi tastați `zmore Serial-HOWTO` pentru a vizualiza fișierul Serial-HOWTO.

Rezumat

Modemurile pun la dispoziția utilizatorului o modalitate convenabilă de apelare și conectare la alte sisteme. De asemenea, puteți configura modemul pentru a permite altor persoane să acceseze sistemul dumneavoastră Linux. Acest capitol descrie modul de utilizare a modemurilor în Linux.

Prin lectura acestui capitol, ați învățat următoarele:

- ▶ În Linux, porturile seriale dispun de două tipuri de fișiere de periferic asociate, în funcție de utilizarea porturilor seriale ca periferice pentru emisia sau recepția de apeluri. Când vă accesați portul serial pentru a emite apeluri prin intermediul unui modem,

numele de periferic sunt `/dev/cua0` și `/dev/cua1` pentru COM1 și COM2. Pe de altă parte, COM1 și COM2 folosesc numele de periferic `/dev/ttyS0`, respectiv `/dev/ttyS1` când sunt utilizate ca periferice pentru recepția apelurilor. Deși versiunile mai vechi de Linux fac diferența între cele două nume de periferic, puteți folosi numele `/dev/ttyS*` atât pentru emiterea, cât și pentru recepția apelurilor.

- ▶ Cipul Universal Asynchronous Receiver Transmitter (UART) reprezintă esența comunicațiilor seriale. Cipul UART convertește baiții emiși în baiți individuali, pe care modemul îi poate converti în formă analogică și îi poate trimite pe liniile telefonice; de asemenea, împachetează baiții recepționați în baiți. Pentru comunicații de mare viteză, aveți nevoie de un cip UART National Semiconductor 16550A sau compatibil. Aceste cipuri UART dispun de buffere de emisie și de recepție, pentru a permite o transmisie fiabilă a datelor în sisteme de operare multitasking cum este Linux, în speță sisteme de operare incapabile de a monitoriza în mod constant porturile seriale.
- ▶ Comunicațiile seriale implică mai mulți parametri, inclusiv rata de baud (sau rata de baiți), lungimea cuvântului, paritatea și numărul de baiți de stop. *Rata de baud* se referă de fapt la numărul de schimbări ale stării liniei de transmisie în unitatea de timp, dar desemnează, de regulă, rata de transmisie a datelor (în baiți pe secundă, sau bps).
- ▶ Standardul RS-232C specifică modul de desfășurare a comunicației seriale între două periferice. RS-232C denumeste modemul ca echipament pentru comunicații de date (Data Communications Equipment - DCE), iar calculatorul este echipament terminal de date (Data Terminal Equipment - DTE). De asemenea, standardul RS-232C mai definește rolul pinilor unui cablu serial.
- ▶ Mai multe standarde definesc modalitatea în care modemurile transmit semnale prin linii telefonice. Uniunea Internațională de Telecomunicații (ITU) a ratificat numeroase standarde de modem, inclusiv V.34 pentru 28800 bps și V.90 pentru 56600 bps (56 K). Pentru a stabili o cale de comunicație, aveți nevoie de două modemuri compatibile situate la cele două extremități ale unei linii.
- ▶ Toate modemurile moderne înțeleg un set de comenzi care încep cu literele AT. Acest set de comenzi AT a fost dezvoltat de Hayes Microcomputer Products, Inc. Puteți configura un modem și efectua apeluri cu acesta folosind apeluri AT.
- ▶ Pentru a executa apeluri în exterior dintr-un sistem Linux, aveți nevoie de un program de telecomunicații. CD-ROM-ul anexat acestei cărți include programul de comunicații `minicom`.
- ▶ Pentru a permite altor persoane să apeleze sistemul dumneavoastră Linux prin intermediul unui modem, trebuie să rulați `ugetty` (sau alte procese `getty`) pe linia serială la care este conectat modemul. De asemenea, trebuie să configurați anumiți parametri în fișierul `/etc/gettydefs`. De regulă, `ugetty` se lansează din fișierul `/etc/inittab`.
- ▶ Puteți conecta un terminal la portul serial, cu condiția să folosiți un cablu corespunzător. De asemenea, Linux suportă numeroase tipuri de plăci seriale I/O multiport, care vă permit să conectați numeroase terminale sau modemuri la sistemul dumneavoastră Linux PC.

Capitolul 16

Rețele

În acest capitol

- ▶ Veți înțelege modelul de rețea pe 7 niveluri OSI
- ▶ Veți înțelege protocoalele de rețea și TCP/IP
- ▶ Veți înțelege adresele de rețea și de gazdă (host) în Internet Protocol (IP)
- ▶ Vă veți familiariza cu serviciile Internet
- ▶ Veți folosi Ethernet cu Linux
- ▶ Veți seta TCP/IP în Linux
- ▶ Veți folosi comenzile de diagnosticare TCP/IP în Linux

Unix și lucrul în rețea merg mână în mână. În particular, TCP/IP este practic sinonim cu UNIX. Ca o clonă a lui UNIX, Linux are încorporate vaste facilități de lucru în rețea. În special Linux are facilități TCP/IP pentru câteva interfețe fizice, ca de exemplu plăci Ethernet, porturi seriale și paralele.

În mod obișnuit, folosiți o placă de rețea Ethernet pentru rețeaua dumneavoastră locală (LAN – Local Area Network) – la serviciu sau chiar la domiciliu (dacă aveți mai multe calculatoare). Utilizarea TCP/IP cu un port serial vă permite să vă conectați la alte rețele prin intermediul unui modem. Linux operează atât cu SLIP (Serial Line Internet Protocol), cât și cu PPP (Point-to-Point Protocol).



Referință încrucișată

Acest capitol se concentrează asupra facilităților din Linux pentru TCP/IP utilizat cu interfața fizică tip Ethernet. Cu toate că multe din aceste chestiuni sunt aplicabile și la TCP/IP utilizat cu port serial, acest capitol nu insistă asupra detaliilor referitoare la rețelele de tip dial-up (prin apel telefonic); acest subiect este tratat în Capitolul 18.

Capitolul începe cu o discuție despre rețele în general și despre TCP/IP în particular, apoi tratează setarea fizică pentru un LAN Ethernet, incluzând informații despre diferite tipuri de plăci Ethernet. În final, capitolul descrie cum trebuie setată o rețea TCP/IP pentru sistemul de operare Linux.



Referință încrucișată

Laptop-urile folosesc deseori plăci de rețea destinate PC-urilor. Capitolul 17 descrie plăcile PC cu care poate lucra Linux.

Bazele lucrului în rețea

Ca orice alt subiect tehnic, și utilizarea rețelelor este un subiect plin de termeni de specialitate și jargon, cuvinte pe care un neinițiat le-ar putea considera descurajatoare. Această secțiune prezintă câteva concepte de bază din domeniul rețelelor, începând cu un model de lucru în rețea pe niveluri și continuând cu detalii despre Ethernet și protocoale de rețea TCP/IP.

Modelul pe șapte niveluri OSI

Un model conceptual foarte răspândit de lucru în rețea este modelul de referință pe șapte niveluri pentru interconectarea sistemelor deschise (OSI – Open Systems Interconnection) conceput de Organizația Internațională de Standardizare (ISO). Modelul de referință OSI descrie fluxul de informații între legătura fizică la rețea și aplicația utilizatorului final. Fiecare nivel este responsabil cu îndeplinirea anumitor funcții, după cum se poate observa în Figura 16-1.

Figura 16-1

Modelul de referință pe șapte niveluri pentru interconectarea sistemelor deschise.

7	Aplicație
6	Prezentare
5	Sesiune
4	Transport
3	Rețea
2	Legătură de date
1	Fizic

După cum se vede în figura 16-1, nivelurile sunt numerotate de jos în sus. Funcțiile de bază, ca de exemplu transmiterea fizică a biților de date prin cablul de rețea, se găsesc în partea de jos, iar funcțiile care se ocupă cu un nivel mai ridicat de abstractizare a datelor se află în partea de sus. Rolul fiecărui nivel este de a furniza servicii nivelului imediat superior astfel încât să nu fie necesar ca nivelul superior să cunoască modul efectiv de implementare a serviciilor.

Funcțiile celor șapte niveluri ale modelului de referință OSI sunt următoarele :

- **Nivelul fizic** (Nivelul 1) transmite biți de date neprelucrați prin intermediul mediului fizic (cablul de rețea sau undele electromagnetice, în cazul rețelelor fără cablu). Acest nivel transportă informația generată de toate nivelurile superioare. Nivelul fizic are trei componente:
 - topologia rețelei (de exemplu magistrală sau stea) care specifică modul de conectare fizică între diferitele noduri ale rețelei
 - mediul de transmitere (de exemplu cablu coaxial RG-58, cablu torsadat ecranat sau neecranat, cablu cu fibre optice și microunde) care transportă semnalele reprezentând informații
 - tehnica transmiterii (de exemplu Carrier Sense Multiple Access with Collision Detection [CSMA/CD] – acces multiplu cu urmărirea purtătoarei și detecția coliziunilor, utilizată de Ethernet; tehnici bazate pe jeton, utilizate de rețele cu topologie Token Ring (jeton în cerc) și de Fiber Distributed Data Interface (FDDI) – interfața pentru date distribuită pe fibre optice), ce definește protocoalele hardware pentru transferul de date.
- **Nivelul de legătură de date** (Nivelul 2) operează cu pachete logice sau cadre de date. Acest nivel „împachetează” biții neprelucrați de pe nivelul fizic în cadre, într-un format care depinde de tipul rețelei, de exemplu Ethernet sau Token Ring. Cadrele folosite de nivelul legătură de date conțin adresele fizice ale expeditorului și destinatarului datelor.
- **Nivelul de rețea** (Nivelul 3) cunoaște adresele logice din rețea și modul de a le obține pe cele fizice, cunoscându-le pe cele logice. La capătul de emisie, nivelul de rețea convertește pachetele logice mari în cadre de date mai mici. La capătul de recepție, nivelul de rețea reasamblează cadrele de date în structura originală de pachete logice.
- **Nivelul de transport** (Nivelul 4) este responsabil pentru livrarea în bune condiții a mesajelor generate pe nivelul de aplicații. La capătul de plecare, acest nivel împarte mesajele lungi în câteva pachete. La capătul de primire, nivelul de transport reasamblează mesajele originale și trimite o confirmare de primire. Nivelul de transport mai verifică dacă informațiile sunt primite în ordinea corectă și într-un mod convenabil. În cazul unor erori, nivelul de transport solicită retransmiterea informațiilor.

■ **Nivelul de sesiune** (Nivelul 5) permite aplicațiilor de pe diferite calculatoare să inițieze, să folosească și să încheie o conexiune (conexiunea este numită sesiune). Nivelul de sesiune convertește numele sistemelor în adrese corespunzătoare acestora (de exemplu adresele IP în rețele TCP/IP).

■ **Nivelul de prezentare** (Nivelul 6) se ocupă de formatul folosit pentru schimbul de date între calculatoarele conectate în rețea. Criptarea și decriptarea informațiilor, de exemplu, se face pe acest nivel. Cele mai multe protocoale de rețea nu au un nivel de prezentare.

■ **Nivelul de aplicație** (Nivelul 7) reprezintă poarta prin care aplicațiile accesează serviciile rețelei. Acest nivel oferă servicii (de exemplu transfer de fișiere, acces la baze de date și poșta electronică) de comunicație directă cu aplicațiile.

Modelul de referință OSI nu este caracteristic pentru orice arhitectură hardware sau software, ci pur și simplu furnizează un cadru arhitectural și pune la dispoziție o terminologie uzuală pentru a discuta despre diferite facilități de lucru în rețea.

Un model TCP/IP simplificat cu patru niveluri de rețea

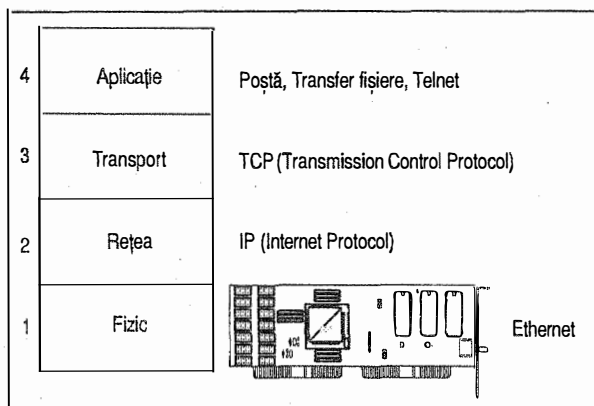
Modelul cu 7 niveluri OSI nu este un model ce trebuie respectat întocmai; el oferă liniile directe pentru organizarea tuturor serviciilor de rețea. Cele mai multe implementări adoptă un model pe niveluri pentru serviciile de rețea, iar aceste modele pot fi reprezentate pornind de la modelul de referință OSI. Modelul TCP/IP de lucru în rețea, de exemplu, poate fi reprezentat printr-un model simplificat.

De obicei, aplicațiile pentru rețele operează cu primele trei niveluri din partea de sus (sesiune, prezentare și aplicație) ale modelului de referință cu șapte niveluri OSI. Astfel, aceste trei niveluri pot fi combinate într-unul singur numit *nivel de aplicație*.

Cele două niveluri din partea de jos a modelului OSI – fizic și legătură de date – pot fi de asemenea combinate într-un singur nivel fizic. Aceste combinații conduc la un model simplificat pe patru niveluri, ca în Figura 16-2.

Figura 16-2

Model simplificat pe patru niveluri de lucru în rețea



Pe fiecare din aceste niveluri, informația este transferată prin intermediul unuia din multitudinea de protocoale de rețea.

Protocoale de rețea

Un *protocol de rețea* desemnează un procedeu detaliat, convenit între emițător și receptor, pentru a schimba informații la un anumit nivel al modelului. Astfel, în Figura 16-2 puteți vedea următoarele protocoale pentru modelul simplificat cu 4 niveluri:

- Protocoale pentru nivelul fizic, de exemplu Ethernet, Token Ring și FDDI.
- Protocoale pentru nivelul de rețea, ca de exemplu IP (Internet Protocol), care face parte din setul de protocoale TCP/IP.
- Pentru nivelul de transport, ca de exemplu protocolul de control al transmisiei (TCP) și protocolul pentru datagrame utilizator (UDP – care face parte din setul de protocoale TCP/IP).
- Protocoale pentru nivelul de aplicație – ca de exemplu protocolul de transfer al fișierelor (FTP), Protocolul de Transfer al Poștei Simple (SMTP), Sistemul pentru denumirile Domeniilor (DNS), Telnet, Protocolul de Transfer al Hipertextului (HTTP) și Protocolul de Management al Rețelei Simple (SNMP) – care fac parte de asemenea din suita de protocoale TCP/IP.

Termenul de *suită de protocoale* se referă la o colecție de două sau mai multe protocoale din aceste niveluri, care formează baza unei rețele. Cele mai cunoscute suite de protocoale sunt următoarele:

- IPX/SPX (Internet Packet Exchange / Sequenced Packet Exchange) folosit de Novell Netware
- NetBIOS și NetBEUI (Network BIOS Extended User Interface – interfață extinsă cu utilizatorul pentru sistem de bază pentru intrări/ieșiri din rețea, folosit de sistemele de operare ale companiei Microsoft)
- TCP/IP



Dintre aceste seturi de protocoale, cel mai important este setul de protocoale TCP/IP, acestea fiind bine suportate de Linux și alte sisteme UNIX.



Pe lângă protocolul TCP/IP, Linux suportă de asemenea protocolul IPX, dar nu și protocolul SPX necesar pentru NetWare. Facilitățile Linux pentru NetBIOS constau într-un pachet de software numit *Samba*, care este inclus pe CD-ROM-ul anexat cărții și descris în secțiunea „Serverul LAN Manager” din Capitolul 21.

Mai multe despre TCP/IP

Acest capitol vă oferă o imagine generală asupra protocolului TCP/IP și rețelelor Ethernet și apoi prezintă instrucțiunile caracteristice Linux pentru setarea rețelelor TCP/IP. Pentru informații suplimentare despre TCP/IP, consultați cărți de specialitate.

- *TCP/IP For Dummies, Second Edition*, IDG Books Worldwide, 1997
- Douglas E. Comer, *Internetworking with TCP/IP, Volume 1, Third Edition*, Prentice-Hall, 1995
- W. Richard Stevens, *UNIX Network Programming, Second Edition*, Prentice-Hall, 1998

- Craig Zacker, *TCP/IP Administration*, IDG Books Worldwide, 1998

TCP/IP și Internetul

TCP/IP a devenit protocolul cel mai folosit pentru Internet – „rețeaua rețelelor” care s-a dezvoltat din ARPAnet, o rețea cu comutație de pachete în care comunicarea între două

calculatoare se realizează fără o setare prealabilă, care s-a dezvoltat în urma cercetărilor realizate de Agenția de Proiecte de Cercetare Avansată (ARPA – Advanced Research Projects Agency) a Guvernului SUA în anii '70. Ulterior, ARPA a luat prefixul *Defense* și a devenit DARPA. Sub auspiciile DARPA, protocoalele TCP/IP au apărut ca o cunoscută colecție de protocoale pentru „internetworking” – termen folosit pentru a descrie comunicarea între rețele.



Observație

TCP/IP s-a dezvoltat din mai multe motive. Un motiv important este acela că protocolul este deschis, ceea ce înseamnă că descrierea lui tehnică apare în documente publice, astfel încât oricine poate alcătui un protocol TCP/IP cu ajutorul pachetului software sau al arhitecturii hardware pe care le are la dispoziție.

Un alt motiv, mai important pentru succesul TCP/IP este disponibilitatea implementării de probă. În loc de a descrie arhitectura rețelei și protocoalele pe hârtie, fiecare componentă a setului de protocoale TCP/IP a fost creată ca o specificație pentru o implementare de probă.

RFC-uri

Detalii referitoare la fiecare protocol TCP/IP sunt descrise în documente cunoscute ca Request for Comments (RFC). Aceste documente sunt distribuite gratuit pe Internet. Le puteți obține de pe următoarele adrese: <http://www.cis.ohio-state.edu/hypertext/information/rfc.html> sau <ftp://venera.isi.edu/in-notes/>.

Chiar și documentația acestei notații folosite pentru a desemna resursele Internet într-un mod uniform poate fi găsită în RFC. Notația, cunoscută ca URL – Uniform Resource Locator (localizator uniform de resurse), este descrisă în RFC 1630, „Universal Resource Identifiers in WWW,” scrisă de T.Berners-Lee, creatorul World Wide Web (WWW).

Puteți considera RFC-urile ca fiind documentele de lucru ale comunității de cercetare și dezvoltare a Internetului. Toate standardele Internet sunt publicate ca RFC-uri. Cu toate acestea, multe RFC-uri nu specifică nici un standard, având caracter informațional.

RFC-uri importante

Iată câteva RFC-uri pe care le-ați putea găsi interesante:

- RFC 768, „User Datagram Protocol (UDP)”
- RFC 791, „Internet Protocol (IP)”
- RFC 792, „Internet Control Message Protocol (ICMP)”
- RFC 793, „Transmission Control Protocol (TCP)”
- RFC 821, „Simple Mail Transfer Protocol (SMTP)”
- RFC 822, „Format for Electronic Mail Messages”
- RFC 950, „IP Subnet Extension”
- RFC 959, „File Transfer Protocol (FTP)”
- RFC 1034, „Domain Names: Concepts and facilities”
- RFC 1058, „Routing Information Protocol (RIP)”
- RFC 1112, „Internet Group Multicast Protocol (IGMP)”

- RFC 1155, „Structure of Management Information (SMI)”
- RFC 1157, „Simple Network Management Protocol (SNMP)”
- RFC 1310, „The Internet Standards Process”
- RFC 1519, „Classless Inter-Domain Routing (CIDR) Assignment and Aggregation Strategy”
- RFC 1521, „Multipurpose Internet Mail Extensions (MIME)”
- RFC 1583, „Open Shortest Path First Routing V2 (OSPF2)”
- RFC 1625, „WAIS over Z39.50-1988”
- RFC 1661, „Point-to-Point Protocol (PPP)”
- RFC 1725, „Post Office Protocol, Version 3 (POP3)”

- RFC 1739, „A Primer on Internet and TCP/IP Tools”
- RFC 1738, „Uniform Resource Locators (URL)”
- RFC 1796, „Not All RFCs are Standards”
- RFC 1855, „Netiquette Guidelines”
- RFC 1866, „Hypertext Markup Language – 2.0”
- RFC 1883, „Internet Protocol, Version 6 (IPv6) Specification”
- RFC 1884, „IP Version 6 Addressing Architecture”
- RFC 1886, „DNS Extensions to support IP version 6”
- RFC 1918, „Address Allocation for Private Internets”
- RFC 2028, „The Organizations Involved in the IETF Standards Process”
- RFC 2060, „Internet Message Access Protocol – Version 4rev1 (IMAP4)”
- RFC 2200, „Internet Official Protocol Standards”
- RFC 2305, „A Simple Mode of Facsimile Using Internet Mail”



Sugestie

RFC-urile continuă să se dezvolte pe măsură ce apar noi tehnici și tehnologii. Dacă lucrați în domeniul rețelor, ar trebui să consultați frecvent RFC-urile pentru a fi la curent cu protocoalele de rețea recent apărute.

Adrese de IP

Când dispuneți de mai multe calculatoare conectate în rețea, aveți nevoie de o modalitate pentru identificarea fiecăruia din aceste calculatoare în mod unic. În rețelele TCP/IP, adresa unui calculator este cunoscută ca *adresă IP*. Deoarece TCP/IP are facilități pentru comunicarea între rețele, adresa este formată din adresa rețelei și adresa calculatorului gazdă, care împreună identifică în mod unic un calculator:

- *Adresa rețelei* indică rețeaua din care face parte calculatorul
- *Adresa calculatorului gazdă* precizează calculatorul din această rețea

Adrese zecimale cu punct

Adresa originală IP este o valoare pe 4 baiți (32 de biți). Convenția este să se scrie fiecare octet ca o valoare zecimală și să se pună un punct (.) după fiecare număr. Astfel, adresa rețelei o puteți vedea scrisă sub forma 140.90.23.100. Această modalitate de a scrie adresele IP este cunoscută ca notație *zecimală cu punct*.

Clase de adrese

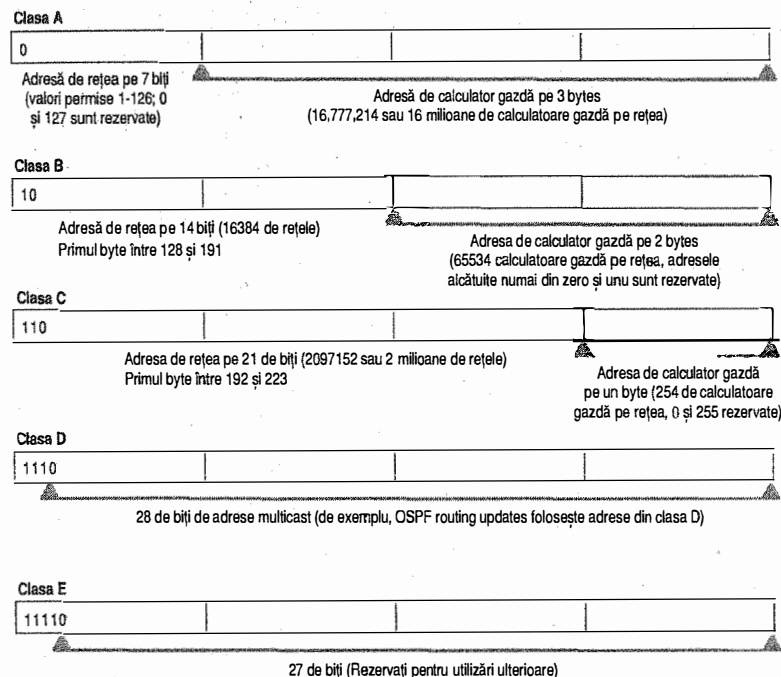
Biții din adresa IP sunt interpretați în felul următor:

<Adresa rețelei, Adresa calculatorului gazdă>

Cu alte cuvinte, un anumit număr de biți din cei 32 ai adresei IP reprezintă adresa unei rețele; restul de biți specifică adresa calculatorului gazdă. Adresa calculatorului gazdă identifică calculatorul dumneavoastră, în timp ce adresa rețelei identifică rețeaua locală (LAN) la care este conectat acesta.

Pentru a fi valabilă în rețele de diferite mărimi (mărimea unei rețele este dată de numărul de calculatoare conectate în rețea), adresa IP include conceptul de clasă de rețea. Există cinci clase de adrese IP, de la A la E, ca în Figura 16-3.

Figura 16-3
Clase de
adrese IP



Din cele cinci clase de adrese, numai clasele A, B și C sunt folosite pentru a desemna adrese de rețele și calculatoare gazdă; clasele de adrese D și E sunt rezervate pentru destinații speciale.

Adresele din clasa A operează cu 126 de rețele, fiecare din ele având până la 16 milioane de calculatoare. Cu toate că adresa rețelei este pe 7 biți, două valori (0 și 127) au semnificații speciale; prin urmare, puteți folosi adrese de clasă A cuprinse numai între 1 și 126.

Adresele din clasa B sunt destinate rețelelor cu până la 65.534 calculatoare. Pot exista, cel mult, 16.384 de rețele clasa B.

Adresele din clasa C sunt pentru firme mici. Fiecare adresă din clasa C permite până la 254 de calculatoare, și pot exista aproximativ 2 milioane de rețele clasa C. Dacă lucrați într-o companie mică, probabil că aveți o adresă de rețea din clasa C.

Puteți deosebi clasa unei adrese IP după primul număr din notarea *zecimală cu punct* astfel :

- Adrese din clasa A: 1.xxx.xxx.xxx până la 126.xxx.xxx.xxx
- Adrese din clasa B: 128.xxx.xxx.xxx până la 191.xxx.xxx.xxx
- Adrese din clasa C: 192.xxx.xxx.xxx până la 223.xxx.xxx.xxx

Cu toate că există cinci clase de adrese IP, următoarele adrese IP au semnificații speciale:

- O adresă cu toți biții zero în porțiunea de rețea indică rețeaua locală – rețeaua unde mesajul cu adresa IP respectivă a fost creat. Astfel, adresa 0.0.0.200 înseamnă numărul 200 al calculatorului gazdă din această rețea clasa C.
- Adresa din clasa A 127.xxx.xxx.xxx este folosită pentru *loopback* – comunicări în cadrul aceluiși calculator gazdă. În mod convențional, 127.0.0.1 este folosită ca adresă de loopback.

Procedurile care trebuie să comunice prin intermediul TCP/IP cu alte proceduri de pe același calculator gazdă folosesc adresa de loopback pentru a evita transmiterea pachetelor de date în afara rețelei.

- Toți biții diferiți de 0 din orice parte a adresei indică un mesaj difuzat. Adresa 128.18.255.255, de exemplu, înseamnă toate calculatoarele din rețeaua de clasa B 128.18. Adresa 255.255.255.255 este cunoscută ca o difuzare limitată; toate stațiile de lucru din rețeaua curentă vor primi pachetul.

Adrese IP pentru rețeaua dumneavoastră

Dacă setați o rețea independentă pe cont propriu, ce urmează a fi conectată la Internet, vă trebuie adrese unice IP pentru rețeaua dumneavoastră. Adresele IP sunt administrate prin intermediul Centrului Internet de Informații pentru Rețele (Internet Network Information Center – InterNIC). În schimbul unei taxe, puteți obține un nume de domeniu (un nume descriptiv pentru rețeaua dumneavoastră cum ar fi, de exemplu, firma_mea.com) și o serie de adrese IP pentru rețeaua dumneavoastră. Pentru a învăța mai multe despre InterNIC și cum să obțineți un nume de domeniu, consultați <http://rs.internic.net/>.

Dacă aveți acces la Internet prin intermediul unui furnizor de servicii Internet (Internet Service Provider – ISP), nu trebuie să vă îngrijoreze procurarea adreselor IP pentru sistemele dvs; furnizorul vă va pune la dispoziție toate adresele IP necesare.

Dacă nu aveți de gând să vă conectați rețeaua la Internet, nu aveți nevoie de nici o adresă IP unică. RFC 1918 („Address Allocation for Private Internets” – alocare de adrese pentru rețele private) vă pune la dispoziție adresele IP pe care le puteți folosi în cadrul rețelelor private (termenul *rețele private* se referă la orice rețea care nu este conectată la Internet). Trei blocuri de adrese IP sunt rezervate pentru rețelele private:

- 10.0.0.0 până la 10.255.255.255
- 172.16.0.0 până la 172.16.255.255
- 192.168.0.0 până la 192.168.255.255

Puteți folosi adrese din aceste blocuri pentru rețeaua dumneavoastră privată fără a fi necesar să vă coordonați cu vreo organizație. De exemplu, eu folosesc adresa din clasa C 192.168.0.0 pentru o mică rețea privată.

Următoarea generație de IP (IPv6)

În momentul în care adresele IP pe 4 baiți au fost create, numărul de adrese părea satisfăcător. Până acum, cel puțin, adresele din clasele A și B devin insuficiente, iar adresele din clasa C se epuizează cu o viteză foarte mare. În plus, răspândirea adreselor din clasa C a introdus o problemă unică. Fiecare adresă din clasa C trebuie să aibă o intrare în tabela de rutare a rețelelor – tabela ce conține informații despre modul de localizare a oricărei rețele în Internet. Prea multe adrese din clasa C înseamnă prea multe intrări în tabelele de rutare.

Organizația Internet Engineering Task Force – IETF a identificat această problemă în 1991, după care a început lucrul la următoarea generație a schemei de adrese IP, numită IPng, al cărei scop era să înlocuiască vechea schemă de adrese pe 4 baiți (numită IPv4, de la IP versiunea 4).



Observație

Au fost propuse și dezbătute numeroase scheme de adrese alternative. Ultimul concurent, cu o adresă de 128 de biți (16 baiți), a fost denumit IPv6 (de la IP versiunea 6). Pe 18 septembrie 1995 IETF a declarat că setul fundamental al protocoalelor de adresare IPv6 să devină standard propus de IETF. Până în prezent

există peste 50 de RFC-uri ce descriu diferite aspecte ale IPv6, de la IPv6 ce operează cu PPP până la transmiterea pachetelor IPv6 ce operează cu Ethernet.

IPv6 a fost conceput pentru a fi cu un pas înainte față de IPv4. Standardul propus furnizează interoperabilitate directă între calculatoarele folosind vechile adrese IPv4 și orice alt calculator ce utilizează IPv6. Ideea este că utilizatorii își pot îmbunătăți performanțele sistemelor astfel încât să poată folosi IPv6 oricând doresc și că operatorii din rețea sunt liberi să-și îmbunătățească performanțele hardware ale rețelei astfel încât să poată folosi IPv6, fără a afecta utilizatorii curenți de IPv4. Implementări de probă ale IPv6 s-au făcut pe mai multe sisteme de operare, inclusiv Linux. Pentru informații suplimentare despre IPv6 în Linux, consultați Linux IPv6 Frequently Asked Questions (FAQ) la adresa <http://www.terra.net/ipv6/>.



Observație

O schemă de adrese pe 128 de biți permite 170,141,183,460,469,232,000,000,000,000,000,000 de calculatoare unice. Ar trebui să ajungă pentru ceva vreme...

Masca rețelei

Masca rețelei este o adresă IP în care biții corespunzători adresei rețelei au valoarea 1, iar ceilalți au valoarea 0. Clasa adresei de rețea determină masca rețelei.

Dacă aveți o adresă din clasa C, de exemplu, masca rețelei este 255.255.255.0. Astfel, rețelele din clasa B au masca de forma 255.255.0.0, iar cele din clasa A au masca de forma 255.0.0.0.

Adresa rețelei

Adresa rețelei este rezultatul operației ȘI pe biți între masca rețelei și orice adresă IP din rețeaua dumneavoastră. Dacă adresa IP a unui sistem din rețeaua dumneavoastră este 206.197.168.200, iar masca rețelei este 255.255.255.0, atunci adresa rețelei este 206.197.168.0. După cum probabil ați observat, adresa rețelei are un zero în zona alocată adresei calculatorului gazdă. Când solicitați un nume de domeniu și o adresă IP din NIC (Network Information Center), obțineți de fapt o adresă de rețea.

Subrețele

Dacă rețeaua dumneavoastră are o adresă din clasa B, atunci primiți un număr de rețea, care poate avea până la 65.534 de calculatoare conectate. Chiar dacă lucrați pentru o supercorporatie care are mii de calculatoare conectate în rețea, este posibil să doriți să împărțiți această rețea în subrețele mai mici. Dacă firma la care lucrați are birouri în imobile diferite, de exemplu, este posibil să doriți ca fiecare birou să facă parte dintr-o rețea separată. Puteți face acest lucru prin preluarea unor biți din porțiunea alocată calculatorului gazdă din adresa IP și alocarea acestora la adresa rețelei. Această procedură este cunoscută ca definirea măștii unei subrețele.

În fond, adăugați mai mulți biți la masca rețelei. Dacă aveți o rețea din clasa B, de exemplu, masca rețelei va fi 255.255.0.0. Dacă vă hotărâți să împărțiți rețeaua în 128 de subrețele, fiecare subrețea având 512 calculatoare, atunci veți folosi 7 biți din spațiul alocat calculatorului gazdă pentru adresa subrețelei. Astfel, masca subrețelei devine 255.255.254.0.

Rutarea în TCP/IP

Noțiunea de rutare se referă la operația de trimitere (forwarding) a informațiilor de la o rețea la alta. Să considerăm cele două rețele de clasa C, 206.197.168.0 și 164.109.10.0. Pentru a transmite pachete de informații de la una din aceste rețele la cealaltă vă trebuie un dispozitiv de rutare.

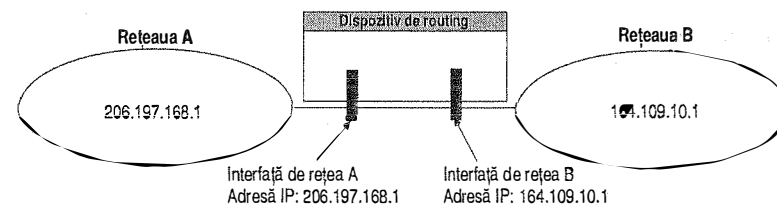


Secret

Deoarece un dispozitiv de rutare facilitează schimbul de date între două rețele, acesta are două conexiuni fizice de rețea, câte una pe fiecare rețea. Fiecare interfață de rețea are propria adresă IP, iar dispozitivul de rutare are rolul principal de a deplasa pachetul de informații înainte și înapoi între cele două interfețe de rețea. Figura 16-4 ilustrează prezența fizică a unui dispozitiv de rutare în două rețele și faptul că fiecare interfață de rețea are propria ei adresă IP.

Figura 16-4

Un dispozitiv de rutare permite schimbul de pachete între două rețele



Termenul generic dispozitiv de rutare se poate referi la un calculator obișnuit cu două interfețe de rețea sau la un dispozitiv proiectat special pentru dirijarea informațiilor. Astfel de instrumente dedicate se numesc *router*.



Observație

Termenul generic *poartă de acces* (gateway) se referă, de asemenea, la orice dispozitiv de rutare. Pentru obținerea unor performanțe bune (rată ridicată de transfer a pachetelor de informații), vă trebuie un router dedicat, al cărui unic scop este dirijarea pachetelor de date într-o rețea.

Mai târziu, după ce veți învăța modul de configurare al unei rețele TCP/IP din Linux, veți întâlni termenul *poartă de acces* (gateway), care se referă la un dispozitiv de rutare, indiferent dacă acesta este un PC sau un router. Trebuie să specificați numai adresa IP pentru poarta de acces în rețeaua dumneavoastră.

Un singur dispozitiv de rutare nu poate conecta toate rețelele din lume; pachetele de date pătrund în Internet trecând dintr-o poartă de acces în alta. Orice rețea conectată cu alta are asociată o poartă de acces. Puteți chiar să asociați anumite porți de acces pentru anumite rețele. După cum veți învăța mai târziu, o tabelă de rutare ține evidența porților de acces asociate cu o rețea externă și a tipului interfeței fizice (ca de exemplu Ethernet sau Point-to-Point Protocol pentru linii seriale) pentru această rețea. Printr-o poartă de acces prestabilită circulă pachete de date adresate oricărei rețele necunoscute.

În rețeaua dumneavoastră locală, toate pachetele de date adresate altei rețele trec prin poarta de acces prestabilită a rețelei dumneavoastră. Dacă această poartă de acces este conectată fizic la rețeaua de destinație, povestea se încheie aici, pentru că poarta de acces poate trimite fizic pachetele de date la destinație. Dacă poarta de acces nu cunoaște rețeaua de destinație, ea trimite pachetele următoarei porți de acces prestabilite (poarta de acces pentru cealaltă rețea pentru care poarta dumneavoastră de acces este disponibilă). Astfel, pachetele de date circulă dintr-o poartă în alta până ajung la rețeaua de destinație (sau până când un mesaj de eroare vă anunță că nu se poate ajunge la aceasta).



Pentru a trimite pachete prin rețea în mod eficient, routerele fac schimb de informații (sub forma tabelelor de rutare) astfel încât fiecare router are o hartă a rețelei în apropierea sa. Routerele schimbă informații folosind un protocol de rute dintr-o familie de protocoale numită Interior Gateway Protocol – IGP. Un IGP foarte folosit este Routing Information Protocol – RIP.

În dirijarea informațiilor TCP/IP, de fiecare dată când un pachet de date trece printr-un router, se consideră că a făcut un *salt*. În RIP, numărul maxim de salturi pentru Internet este de 15. Dacă, plecând de la rețeaua dumneavoastră, un pachet de date nu ajunge la rețeaua de destinație după ce a făcut 15 salturi se consideră că rețeaua respectivă este intangibilă din rețeaua dumneavoastră. Cu alte cuvinte, nu se poate ajunge la o rețea aflată la o depărtare mai mare de 15 routere.

În cadrul unei rețele individuale, nu aveți nevoie de un router dacă nu folosiți o mască de subrețea ca să împărțiți rețeaua individuală IP în mai multe subrețele. Dacă folosiți o mască de subrețea, este necesar să setați routerele pentru a trimite pachete de date dintr-o subrețea în alta.

Sistemul de nume ale domeniilor

Puteți accesa orice calculator gazdă dintr-o rețea TCP/IP dacă îi cunoașteți adresa IP. Memorarea adreselor IP, chiar numai pentru câteva calculatoare gazdă de interes, este incomodă. Acest fapt a fost recunoscut de la începutul protocolului TCP/IP și ca urmare s-a folosit asocierea dintre un calculator gazdă și o adresă IP. Conceptul este asemănător cu cel al unei cărți de telefon, în care căutați un număr după numele persoanei.

La început, asocierea între nume și adresele IP era păstrată într-un fișier text numit HOSTS.TXT din cadrul Centrului de Informații pentru Rețele (NIC), localizat la Institutul de Cercetări Stanford. Acest fișier conținea numele și adresele IP corespunzătoare pentru rețele, calculatoarele gazdă și routerele din Internet. Toate calculatoarele gazdă din Internet transferau acest fișier folosind protocolul FTP. (Vă puteți imagina toate calculatoarele gazdă accesând un fișier dintr-o singură sursă în Internetul de astăzi?). Numărul calculatoarelor gazdă fiind în creștere, ideea unui singur fișier a devenit de neconceput. Fișierul calculatoarelor gazdă devenea greu de întreținut, și era greu pentru toate calculatoarele gazdă să țină la zi acest fișier într-un mod convenabil. Pentru a ușura problema, RFC-urile 881, 882 și 883 au introdus conceptul și planurile pentru numele de domenii în noiembrie 1983. În cele din urmă, aceasta a condus la sistemul de nume ale domeniilor (DNS), cum îl știm astăzi (documentația pentru DNS se găsește în RFC-urile 1032, 1033, 1034 și 1035).

Ierarhia numelor de domenii

DNS furnizează un sistem ierarhic de nume foarte asemănător cu adresa poștală, pe care o puteți citi ca „numele”, „stradă”, „orașul” și „țara”. Dacă aș cunoaște adresa dumneavoastră poștală completă, v-aș putea localiza începând cu orașul și țara în care locuiți. Apoi aș localiza strada și locuința dumneavoastră, aș suna la ușă și aș întreba de dumneavoastră cunoscându-vă numele.



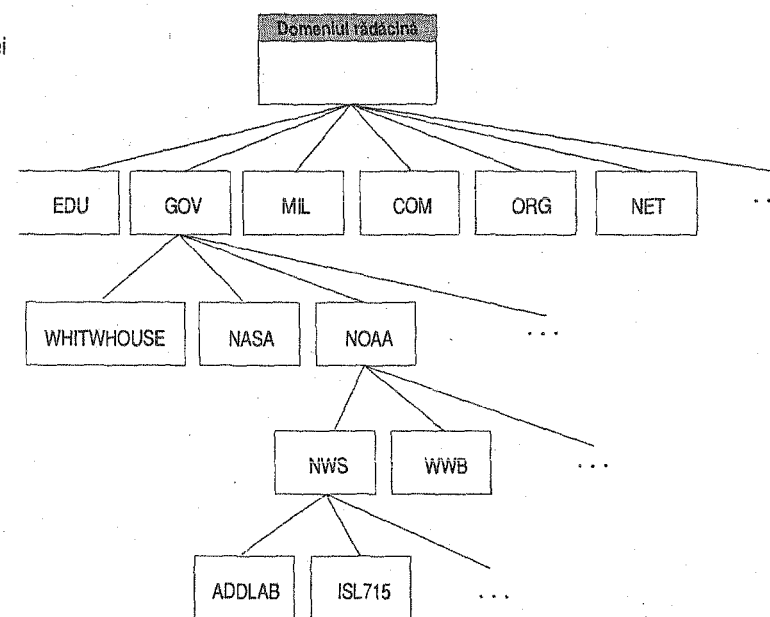
Observație

DNS furnizează, în esență, o schemă de definire a adresei unui calculator gazdă de pe Internet foarte asemănătoare cu adresa poștală. Întregul Internet este împărțit în câteva domenii, ca de exemplu gov, edu, com, mil și net. Fiecare domeniu este împărțit la rândul lui în subdomenii. În cele din urmă, în cadrul unui subdomeniu, fiecărui calculator gazdă îi este dat un nume simbolic. Pentru a scrie numele complet al domeniului, trebuie scrise într-un singur cuvânt numele calculatorului gazdă, numele subdomeniului și numele domeniului folosind punctele (.) ca separatori. Iată care este numele complet al domeniului unui calculator gazdă numit ADDLAB din subdomeniul NWS, care la rândul lui face parte din subdomeniul NOAA al domeniului GOV: ADDLAB.NWS.NOAA.GOV. Rețineți că numele domeniului nu face distincția între literele mari și cele mici.

Figura 16-5 ilustrează o parte a sistemului numelor de domenii din Internet, în speță locația calculatorului gazdă ADDLAB.NWS.NOAA.GOV.

Figura 16-5

O parte a ierarhiei denumirii domeniilor din Internet.



Pentru un sistem comercial din domeniul COM, numele unui calculator gazdă poate fi foarte simplu, ca de exemplu METROLINK.COM.



Sugestie

Vă puteți referi la un utilizator de pe un sistem adăugând semnul „at” (@), urmat de numele domeniului calculatorului gazdă, la numele utilizatorului (numele sub care utilizatorul se conectează în rețea). Astfel, vă puteți referi la vânzările (sales) utilizatorului care are adresa calculatorului gazdă METROLINK.COM astfel: sales@METROLINK.COM (spre deosebire de numele calculatorului gazdă, numele utilizatorului face distincția între literele mari și cele mici).

În acest fel precizați utilizatorii atunci când trimiteți e-mail.

Servere de nume

Aplicațiile pentru rețele TCP/IP convertesc numele unui calculator gazdă într-o adresă IP consultând un *server de nume*, care este alt calculator gazdă accesibil din rețeaua dumneavoastră. Dacă vă hotărâți să folosiți DNS în rețeaua dumneavoastră, trebuie să setați un server de nume sau să indicați unul (printr-o adresă IP).

Secțiunile ce urmează ale acestui capitol tratează fișierele de configurare /etc/host.conf și /etc/resolv.conf, cu ajutorul cărora specificați modul în care numele calculatoarelor gazdă sunt convertite în adrese IP. În particular, specificați adresele IP ale serverului de nume în fișierul /etc/resolv.conf.

Dacă nu folosiți DNS, puteți avea o mapare între numele calculatorului gazdă și adresele IP prin intermediul unui fișier text numit /etc/hosts. Intrările într-un fișier tipic /etc/hosts pot arăta ca în exemplul următor:

```
# Acesta este un comentariu
127.0.0.1 localhost
```

```
206.197.168.2 lnbsun
206.197.168.50 lnbs386
206.197.168.100 lnbs486
206.197.168.150 lnbsmac
206.197.168.200 lnbssoft
```

După cum se vede în exemplu, fișierul listează un nume de calculator gazdă pentru fiecare adresă IP. Adresa IP și numele calculatorului gazdă vor fi, desigur, diferite pentru sistemul dumneavoastră.



Secret

Dacă se folosește fișierul `/etc/hosts` pentru căutarea numelor, o problemă o reprezintă faptul că trebuie să copiați acest fișier pe fiecare sistem din rețeaua dumneavoastră. Această procedură poate deveni incomodă chiar pentru o rețea cu numai cinci sau șase sisteme.

Servicii TCP/IP și arhitectura client/server

Din proiectare, un serviciu tipic pentru Internet este implementat în două părți – un server care furnizează informații și unul sau mai mulți clienți care solicită informații. O astfel de arhitectură client/server și-a câștigat popularitatea ca modalitate de implementare a sistemelor de informații distribuite. Arhitectura client/server tipică constă dintr-o colecție de calculatoare conectate printr-o rețea de comunicare. Atribuțiile sistemului de informații sunt realizate de proceduri (programe de calculator) care rulează pe aceste calculatoare și comunică prin rețea.

În ultimii ani, arhitectura client/server a devenit mecanismul obișnuit cu ajutorul căruia bazele de date colective centralizate pot fi accesate de calculatoarele dintr-o rețea. Într-un mediu de lucru client/server, unul sau mai multe servere administrează baza de date centralizată, iar clienții accesează datele prin intermediul serverului.

La fel ca un server de baze de date, un serviciu Internet, precum FTP sau un browser Web, furnizează un serviciu folosind modelul client/server. Un utilizator care dorește să acceseze o informație folosește un client (de exemplu un browser de Web) ca să se conecteze la un server și să copieze informația (de exemplu pagini Web dintr-un server Web). În acest caz, serverul Web se comportă ca un administrator de baze de date – datele sunt fișiere HTML (pagini de Web).



Observație

Arhitectura client/server are nevoie de clienți pentru a comunica cu serverele. Aici intervine rolul protocolului TCP/IP, care furnizează un standard pentru clienți și servere pentru a schimba date. Următoarele secțiuni explică cum comunică între ele serviciile bazate pe TCP/IP. Cu această ocazie veți învăța despre numele porturilor asociate cu binecunoscute servicii Internet.

TCP/IP și Sockets

Aplicațiile client/server, ca de exemplu serverele Web și browserele, folosesc protocoalele TCP/IP pentru a comunica. Aceste aplicații Internet realizează comunicații TCP/IP folosind interfața Berkeley Sockets (denumită astfel pentru că interfața Sockets a fost introdusă în Berkeley UNIX în anul 1982). Interfața Sockets este alcătuită dintr-o bibliotecă de rutine pe care un dezvoltator le poate folosi pentru a crea aplicații ce pot comunica cu alte aplicații din Internet. Există chiar și o Windows Sockets API (interfață de programare a aplicațiilor Windows Sockets – un nume excentric pentru bibliotecă de funcții) care este modelată după interfața Berkeley Sockets. Interfața Winsock, așa cum este cunoscută, furnizează un standard API pe care programatorii Windows îl pot folosi pentru a scrie aplicații de rețea.



Observație

Chiar dacă nu scrieți aplicații de rețea folosind soclu, trebuie să folosiți multe aplicații de rețea. Cunoașterea noțiunii de socket (soclu) vă poate ajuta să înțelegeți cum funcționează aplicațiile pentru rețele, care la rândul lor vă permite să depistați și să corectați orice problemă care poate surveni în aceste aplicații.

Definiția unui socket

Aplicațiile de rețea folosesc socluri pentru a comunica într-o rețea TCP/IP. Un socket este o entitate abstractă care reprezintă capătul bidirecțional al unei aplicații. Deoarece un socket este bidirecțional, datele pot fi trimise și primite. Un socket (soclu) are trei atribute:

- Adresa de rețea (adresa IP) a sistemului
- Numărul portului care identifică procesul (un proces este un program ce rulează pe un computer) ce schimbă informații prin intermediul soclului.
- Tipul de soclu (ca de exemplu un flux sau o datagramă) care identifică protocolul pentru schimbul de date.

În esență, adresa IP identifică un nod de rețea, numărul portului identifică o procedură pe acel nod, iar tipul soclului determină modalitatea de transfer al datelor – printr-un protocol cu conexiune sau fără conexiune.

Protocoalele cu conexiune

Tipul soclului indică protocolul folosit pentru comunicarea prin intermediul acestuia. Un protocol cu conexiune este similar cu o conversație obișnuită la telefon. Când doriți să vorbiți cu un prieten, trebuie mai întâi să formați numărul de telefon al acestuia și să stabiliți o legătură înainte de a putea conversa. În același mod, schimbul de date folosind o conexiune necesită atât trimiterea, cât și primirea de procese pentru a stabili o conexiune înainte ca schimbul de date să înceapă.



Observație

În suita de protocoale TCP/IP, TCP – protocol de control al transmisiei – operează cu transfer de date cu conexiune între două procese ce rulează pe două calculatoare din Internet. TCP furnizează un schimb de date sigur, bidirecțional între procese.

Așa cum sugerează numele TCP/IP (și după cum se vede în secțiunea „Protocoale de rețea”), TCP se bazează pe IP – protocol Internet – pentru transmiterea pachetelor. IP nu garantează livrarea pachetelor și nici nu livrează pachetele într-o anumită ordine. IP doar transmite eficient pachetele dintr-o rețea în alta. TCP este responsabil pentru aranjarea pachetelor în ordinea corectă, pentru detectarea erorilor survenite și solicitarea retransmiterii pachetelor în caz de eroare.

TCP este folosit pentru aplicații care intenționează să transfere mari cantități de date la un moment dat. De asemenea, aplicațiile care au nevoie de un schimb de date sigur folosesc TCP. De exemplu, FTP folosește TCP pentru a transfera fișiere.

În modelul cu socluri, un soclu care folosește TCP este denumit *soclu de flux*.

Protocoale fără conexiune

Un protocol pentru schimbul de date fără conexiune nu solicită ca emițătorul și receptorul să stabilească în mod explicit o conexiune. Este ca și cum ți-ai striga prietenul într-o cameră aglomerată; niciodată nu poți fi sigur că te-a auzit.



Observație

În suita de protocoale TCP/IP, protocolul datagramelor utilizator (UDP) furnizează servicii fără conexiune pentru emiterea și recepția pachetelor cunoscute sub numele de *datagramă*. Spre deosebire de TCP, UDP nu garantează că datagramele ajung la destinație și nici că ele au fost livrate în ordinea în care au fost trimise.

UDP este folosit de aplicații care transferă cantități mici de informații la un moment dat sau de aplicații care nu necesită o siguranță sau o ordine de transmitere a datelor. De exemplu, protocolul SNMP folosește UDP pentru a transfera date.

În modelul cu socluri, un soclu care folosește UDP este denumit *soclu de datagramă*.

Socluri și modelul client/server

Pentru a stabili o cale de comunicație sunt necesare două socluri. Când două procese comunică, ele folosesc modelul client/server pentru stabilirea conexiunii. Aplicația server ascultă la un anumit port din sistem – serverul este identificat complet de adresa IP a sistemului pe care rulează și de numărul portului pe care-l folosește pentru conexiuni. Clientul inițiază conexiunea de pe orice port disponibil și încearcă să se conecteze la server (identificat de adresa IP și numărul portului). O dată conexiunea stabilită, clientul și serverul pot schimba date în conformitate cu propriul lor protocol.

Succesiunea evenimentelor în schimbul de date bazat pe socluri depinde tipul de transfer: dacă acesta este cu conexiune (TCP) sau fără (UDP).

Pentru un transfer de date cu conexiune folosind socluri, serverul „ascultă” un anumit port, așteptând clienții să solicite conectarea. Transferul de date începe numai după ce a fost stabilită o legătură.

Pentru transferuri de date fără conexiune, serverul așteaptă ca o datagramă să sosească la portul specificat. Clientul nu așteaptă să se stabilească o legătură; el pur și simplu trimite o datagramă către server.

Comunicații client/server cu TCP/IP

Aplicațiile client/server folosesc următoarele etape de bază pentru a efectua schimburi de date într-o rețea TCP/IP:

1. Creează un soclu.
2. Asociază de soclu o adresă IP și un port.
3. Ascultă conexiunile, dacă aplicația este un server ce folosește un soclu de flux.
4. Stabilește conexiunea, dacă aplicația este un client ce folosește un soclu de flux.
5. Execută schimbul de date.
6. Închide soclul când schimbul de date a luat sfârșit.

Pentru socluri fără conexiune (care implementează transferul de date folosind UDP) trebuie omise etapele 3 și 4.

Indiferent dacă este un server sau un client, fiecare aplicație creează mai întâi un soclu. Apoi asociază soclul cu adresa IP a calculatorului local și cu un număr de port. Adresa IP identifică mașina (unde rulează aplicația) iar numărul portului identifică aplicația care folosește soclul.



Observație

Serverele, în mod obișnuit, folosesc un număr de port dedicat (well-known), astfel încât clienții să se poată conecta la acel port pentru a accesa serverul. Pentru o aplicație client, procedeu de legare a unui soclu de o adresă IP și de un port este identic cu cel pentru o aplicație server, dar clientul poate folosi zero ca număr de port – biblioteca de socluri folosește automat pentru client un număr de port nefolosit.

Pentru un soclu de flux cu conexiune, aplicațiile ce comunică între client și server trebuie să stabilească o conexiune. Etapele exacte pentru stabilirea unei conexiuni depind de natura aplicației: server sau client.



Observație

În modelul client/server, serverul trebuie să fie în stare de funcționare înainte de a putea clientul să ruleze. După crearea unui soclu și asocierea lui de un port, aplicația server creează o coadă a legăturilor, care determină câți clienți se pot conecta la un server. În mod obișnuit, un server poate folosi până la 5 conexiuni. În orice caz, mărimea cozii de ascultare este unul din parametrii pe care îi puteți ajusta (în special pentru un server Web) ca să vă asigurați că serverul răspunde unui număr maxim de clienți.

După crearea cozii de ascultare, serverul așteaptă o conexiune cu un client.

Stabilirea unei legături de către client este ceva mai simplă. După crearea unui soclu și asocierea acestuia la o adresă de rețea, clientul stabilește conexiunea cu serverul. Pentru a face conexiunea, clientul trebuie să știe numele rețelei sau adresa IP a serverului, precum și portul pe care serverul acceptă conexiunea. După cum o arată și secțiunea următoare, toate serviciile Internet au numere de porturi standard dedicate.



Observație

După ce un client stabilește o legătură la un server folosind un soclu de flux cu conexiune, clientul și serverul pot schimba date prin apelarea funcțiilor soclu API corespunzătoare. Ca într-o conversație între două persoane, serverul și clientul trimit și primesc date alternativ – semnificația datelor depinde de protocolul de mesaj folosit de server și clienți. De obicei, un server este proiectat pentru o anumită sarcină și bineînțeles dispune și de un protocol de mesaj pe care serverul și clienții îl folosesc pentru a schimba informațiile necesare. De exemplu, serverul Web și browserul Web (clientul) comunică folosind protocolul HTTP (de transfer prin hipertext).

Servicii Internet și numere de porturi

Suita de protocoale TCP/IP a devenit limbajul universal pentru Internet, deoarece multe servicii standard sunt disponibile pe toate sistemele care operează cu TCP/IP. Aceste servicii sunt punctele forte ale Internetului, facilitând transmiterea poștei, a știrilor și a paginilor Web. Aceste servicii poartă nume binecunoscute, cum sunt următoarele:

- FTP (File Transfer Protocol – Protocol de transfer al fișierelor) permite transferul fișierelor între calculatoarele din Internet. FTP folosește două porturi – datele sunt transferate pe portul 20, în timp ce schimbul informațiilor de control se realizează la 21.
- HTTP (Hypertext Transfer Protocol – Protocol de transfer prin hipertext) este un protocol recent de trimitere a documentelor HTML de pe un sistem pe altul. HTTP este protocolul fundamental pentru Web. În mod prestabilit, serverul Web și clientul comunică pe portul 80.
- SMTP (Simple Mail Transfer Protocol – Protocol pentru transferul de mesaje simple) pentru schimbul de mesaje e-mail între sisteme. SMTP folosește portul 25 pentru schimbul de informații.
- NNTP (Network News Transfer Protocol – Protocol pentru transferul de mesaje în rețea) pentru distribuirea articolelor cu noutăți pe Internet în maniera stochează-și-trimite. NNTP folosește portul 119.
- Telnet permite utilizatorilor dintr-un sistem să se conecteze la alt sistem de pe Internet (utilizatorul trebuie să introducă un identificator și o parolă validă pentru a se conecta la sistemul aflat la distanță). În mod prestabilit, Telnet folosește portul 23. Totuși, clientul Telnet se poate conecta la orice port specificat.

■ **SNMP** (Simple Network Management Protocol – Protocol pentru managementul unei rețele simple) pentru administrarea tuturor tipurilor de dispozitive de rețea de pe Internet. Ca și FTP, SNMP folosește două porturi: 161 și 162.

■ **TFTP** (Trivial File Transfer Protocol – protocol obișnuit pentru transferul fișierelor) pentru transferarea fișierelor de la un sistem la altul (folosite în mod obișnuit de terminale X și de stații de lucru fără hard-disc pentru a descărca fișiere de inițializare de pe alt calculator gazdă din rețea). Transferul de date TFTP se face pe portul 69.

■ **NFS** (Network File System – sistem de fișiere de rețea) pentru partajarea fișierelor între calculatoare. NFS folosește facilitatea RPC de la Sun, care schimbă informațiile pe portul 111.

■ **WAIS** (Wide Area Information Service – serviciu de informații pe suprafețe mari) – pentru a căuta date (informații) pe sistemele distribuite. WAIS folosește portul 210 (în fișierul /etc/services, veți găsi informații despre WAIS sub rubrica denumită z3950 pentru că WAIS folosește protocolul standard ANSI Z39.50)

Un port dedicat este asociat cu fiecare dintre aceste servicii. Protocolul TCP folosește acest port pentru a localiza un serviciu pe orice sistem. (Un proces de server – un program care rulează pe un sistem – implementează fiecare serviciu.)

Ca și fișierul /etc/hosts, care stochează asocierea dintre numele calculatorului gazdă și adresele lor IP, asocierea dintre numele unui serviciu și numărul unui port (precum și al unui protocol) este stocată în alt fișier text, numit /etc/services.

În continuare, sunt prezentate câteva intrări din fișierul /etc/services al unui sistem Linux:

```
ftp-data 20/tcp
ftp      21/tcp
telnet   23/tcp
smtp     25/tcp      mail
time     37/tcp      timserver
time     37/udp      timserver
rlp      39/udp      resource # locatia resursei
name     42/udp      nameserver
ehois    43/tcp      nickname # de obicei la sri-nic
domain   53/tcp
domain   53/udp
mtp      57/tcp      # aspru criticat
bootps   67/udp      # bootp server
bootpc   68/udp      # bootp client
tftp     69/udp
gopher   70/tcp      # server gopher
rje      77/tcp
finger   79/tcp
http     80/tcp      # www este folosit de unele programe defecte,
www      80/tcp      # http este mai corect
link     87/tcp      ttylink
kerberos 88/udp      kdc # verificarea identitatii kerberos --udp
kerberos 88/tcp      kdc # verificarea identitatii kerberos -- tcp
```

Veți considera interesantă parcurgerea intrărilor din fișierul /etc/services, deoarece ele prezintă multitudinea serviciilor de rețea disponibile sub protocolul TCP/IP.



Observație

Rețineți că portul numărul 80 este proiectat pentru WWW. Dacă setați un server Web pe sistemul dumneavoastră, acel server ascultă la portul 80.

Superserverul inetd

Modelul client/server necesită ca serverul să fie pornit și să ruleze anterior emiterii unei cereri de servicii de către client. O idee simplistă ar fi aceea de a ține pornite toate serverele, tot timpul. Totuși, această idee nu este practică, pentru că fiecare proces de server ar utiliza în mod excesiv resurse de sistem sub formă de memorie și timp de procesare. Mai mult decât atât, nu aveți nevoie ca toate serviciile să fie gata în orice moment. O soluție inteligentă pentru această problemă este rularea unui singur server **inetd**, care ascultă de toate porturile și pornește serverul corespunzător atunci când intervine solicitarea unui client.

De exemplu, când un client încearcă să se conecteze la portul FTP, **inetd** pornește serverul FTP și îi permite acestuia să comunice direct cu clientul (iar serverul FTP devine inactiv atunci când clientul se deconectează).

Deoarece pornește diverse servere la cerere, **inetd** este recunoscut ca superserverul Internetului.

În mod obișnuit, un sistem UNIX pornește **inetd** când sistemul se inițializează. Serverul **inetd** citește la inițializare un fișier de configurare numit /etc/inetd.conf.

Acest fișier specifică serverului **inetd** porturile la care să asculte și serverul care trebuie activat pentru fiecare port. De exemplu, pentru sistemul meu intrarea în fișierul /etc/inetd.conf care pornește serverul FTP arată astfel:

```
ftp      stream tcp      nowait root      /usr/sbin/tcpd in.ftpd -l -a
```

Primul articol din această linie, FTP, indică serverului **inetd** numele serviciului. **inetd** folosește acest nume pentru a căuta numărul portului în fișierul /etc/services. Dacă folosiți comanda **grep** pentru a căuta ftp în fișierul /etc/services, iată ce veți găsi:

```
grep ftp /etc/services
ftp-data 20/tcp
ftp      21/tcp
tftp     69/udp
sftp     115/tcp
```

Din rândurile de mai sus puteți observa că numărul portului pentru serviciul FTP este 21.

Acesta spune serverului **inetd** să asculte la portul 21 pentru solicitări de servicii FTP.

Restul câmpurilor din intrarea FTP au următoarele semnificații:

- Al doilea și al treilea câmp, **stream tcp**, spun serverului **inetd** că serviciul FTP folosește un soclu TCP orientat pe conexiune pentru a comunica cu clientul. Pentru servicii ce folosesc socluri UDP fără conexiuni, aceste două câmpuri sunt dgram udp.
- Al patrulea câmp, **nowait**, spune serverului **inetd** să pornească un server nou pentru fiecare solicitare. Dacă acest câmp are valoarea **wait**, **inetd** așteaptă până când serverul este închis, pentru a-l porni din nou.
- Al cincilea câmp furnizează identificatorul utilizatorului folosit de **inetd** pentru a porni serverul. În acest caz, serverul pornește serverul FTP ca **root** (rădăcina).
- Al șaselea câmp specifică programul ce trebuie rulat pentru acest serviciu, iar ultimul câmp este argumentul pe care **inetd** îl furnizează programului server. În acest caz, programului /usr/sbin/tcpd îi este furnizat argumentul in.ftpd.



Secret

Observați că programul /usr/bin/tcpd este o facilitare de control a accesului pentru servicii Internet. Programul **tcpd** poate porni alte servicii, ca de exemplu FTP și Telnet, dar, înainte de a porni serviciul, **tcpd** consultă fișierul /etc/hosts.allow pentru a vedea dacă calculatorului gazdă îi este permis serviciul pe care-l solicită.

Dacă în fișierul `/etc/hosts.allow` nu este menționat nimic despre acel calculator gazdă, `tcpd` verifică fișierul `/etc/hosts.deny` pentru a vedea dacă serviciul trebuie interzis. Dacă ambele fișiere sunt goale, `tcpd` permite calculatorului gazdă accesul la serviciul solicitat. Puteți introduce linia `ALL:ALL` în fișierul `/etc/hosts.deny` pentru a interzice accesul tuturor calculatoarelor gazdă la orice servicii Internet (citiți secțiunile „`/etc/hosts.allow`” și „`/etc/hosts.deny`” pentru a învăța mai multe despre aceste fișiere de control al accesului).

Parcurgeți fișierul `/etc/inetd.conf` de pe sistemul dumneavoastră pentru a afla tipurile de servicii pentru care `inetd` este setat să pornească. Câteva dintre aceste servicii (ca de exemplu `finger`, `systat` și `netstat`) furnizează informații ce pot fi folosite de intruși pentru a pătrunde în sistemul dumneavoastră. Puteți dezactiva aceste servicii prin plasarea unui caracter de comentariu (`#`) la începutul liniilor care pornesc aceste servicii. Când faceți o astfel de modificare în fișierul `/etc/inetd.conf`, trebuie să porniți din nou serverul `inetd` parcurgând următorii pași:

1. Folosiți comanda `ps` cu opțiunile corespunzătoare și determinați identificatorul de proces (ID) pentru serverul `inetd`. De exemplu, în Linux, tastați `ps x | grep inetd` și observați primul număr din linia care rezultă. Acesta va fi identificatorul de proces al serverului `inetd`.
2. Tastați `kill -HUP pid`, unde `pid` este identificatorul procesului (un număr) pentru `inetd`. Această comandă repornește serverul `inetd`, care citește din nou fișierul `/etc/inetd.conf`.

Servere autonome

Cu toate că pornirea serverelor prin intermediul `inetd` este o soluție inteligentă, `inetd` nu este suficient dacă un serviciu trebuie pornit frecvent. Serverul Web, în mod obișnuit, trebuie pornit frecvent pentru că de fiecare dată când un utilizator execută clic pe o legătură dintr-o pagină Web, la serverul Web ajunge o cerere. Pentru astfel de servicii pretențioase, este cel mai bine să porniți serverul individual. Astfel de servere de sine stătătoare sunt proiectate să ruleze ca *daemoni* – procese care rulează continuu. Serverul ascultă de portul ce-i este atribuit și tratează cererile care sosesc prin autocopiere. Astfel, serverul funcționează continuu. O strategie mai eficientă, folosită pentru serverele Web, este de a rula copii multiple ale serverului și de a lăsa fiecare copie să trateze câteva dintre cererile primite.

Ethernet și Linux

TCP/IP este o suită de protocoale adecvată pentru rețele, dar înainte de a o utiliza trebuie să configurați rețeaua fizică. Ethernet este o alegere bună pentru mecanismul de transport fizic al datelor din următoarele motive:

- Ethernet este o tehnologie îndelung experimentată (este folosită încă din anii 1980).
- Ethernet furnizează viteze mari de transfer de date: în mod obișnuit 10 milioane de biți pe secundă (10 Mbps – 10 megabiți pe secundă) cu toate că acum există Ethernet cu 100 Mbps.
- Componentele hardware pentru Ethernet au un cost relativ scăzut (plăcile Ethernet pentru PC-uri costă mai puțin de 100 de dolari).

În mod obișnuit, Linux lucrează cu componente hardware accesibile ca preț, iar Ethernet nu face excepție. Următoarele secțiuni descriu plăcile Ethernet cu care poate lucra Linux și setările fizice ale unei rețele locale Ethernet.

Generalități despre Ethernet

Ethernet reprezintă o modalitate standard de a deplasa pachete de date între două sau mai multe calculatoare conectate la un singur cablu. (Rețelele mai mari sunt construite prin conectarea

mai multor segmente Ethernet cu porți de acces.) Deoarece este folosit un singur fir, trebuie folosit un protocol pentru trimiterea și primirea datelor, pentru că la un moment dat poate exista pe cablu doar un singur pachet de date. O rețea locală Ethernet folosește un protocol de transmitere a datelor cunoscut ca *Carrier Sense Multiple Access/Collision Detection* – (CSMA/CD) pentru a facilita folosirea unui singur cablu de transmitere de către mai multe calculatoare. Controllerele Ethernet înglobate în calculatoare urmăresc ca protocolul CSMA/CD să transmită și să primească pachete Ethernet.

Modul de lucru al protocolului CSMA/CD este asemănător cu felul în care purtați o conversație la o petrecere. Așteptați o pauză (dectecția purtătoarei) și vorbiți atunci când toată lumea tace. Dacă dumneavoastră și altă persoană începeți să vorbiți în același timp, vă dați seama imediat de problema apărută (dectecția coliziunii) și vă opriți pentru un moment; apoi unul dintre dumneavoastră începe să vorbească din nou. După cum știți din experiență, totul merge bine.

Într-o rețea locală Ethernet, fiecare controller Ethernet verifică dacă pe cablu sunt prezente semnale – această acțiune reprezintă dectecția purtătoarei. Dacă nivelul semnalului este scăzut, un controller trimite pachetele lui pe cablu. Pachetul conține informații despre expeditor și locul unde trebuie trimis. Toate controllerele Ethernet din rețeaua locală ascultă semnalul, iar receptorul primește pachetul. Dacă se întâmplă ca două controliere să trimită un pachet simultan, nivelul semnalului în cablu crește deasupra unui prag, iar controllerele știu că a survenit o coliziune (două pachete au fost trimise în același timp). Ambele controllere așteaptă o perioadă de timp aleatoare și trimit pachetele din nou.

Ethernetul a fost inventat în anii 1970 la Centrul de Cercetare Xerox Palo Alto (PARC) de Robert M. Metcalfe. În anii '80, Ethernet-ul a fost standardizat prin efortul comun a trei companii: Digital Equipment Corporation (DEC), Intel și Xerox. Folosind inițialele numelor acestor companii, standardul Ethernet a devenit cunoscut ca *standardul DIX*. Mai târziu, standardul DIX a fost inclus în seriile de standarde 802 dezvoltate de Institutul de inginerie electrotehnică și electronică (Institute of Electrical and Electronics Engineers – IEEE). Denumirea oficială pentru Ethernet este IEEE 802.3 CSMA/CD, însă oamenii continuă să-l numească Ethernet.

Ethernet-ul trimite datele în pachete (cunoscute de asemenea și sub denumirea de cadre) având un format standard ce constă din următoarea succesiune de componente:

- 8 biți introducere
- 6 biți adresa de destinație
- 6 biți adresa sursă
- 2 biți lungimea câmpului de date
- biții de la 46 la 1 și 500 – câmpul de date
- 4 biți secvența de verificare a cadrelor (folosită pentru verificarea erorilor).

Nu este necesar să știți prea mult despre mecanismul pachetelor Ethernet, doar să observați adresele sursă și destinație pe 6 biți. Fiecare controller Ethernet are o adresă unică pe 6 biți (48 de biți). La nivelul fizic, pentru aceste pachete trebuie folosite aceste adrese pe 6 biți.

Protocolul de rezoluție al adresei

Într-o rețea locală Ethernet, două controllere Ethernet pot comunica numai dacă fiecare știe adresa fizică Ethernet pe 6 biți a celuilalt. Vă întrebați cum se realizează legătura între adresele IP și adresele fizice. Această problemă este rezolvată de protocolul de rezoluție al adresei (Address Resolution Protocol – ARP), care specifică modul de obținere al adresei fizice

corespunzătoare unei adrese IP. În esență, când un pachet trebuie trimis la o adresă IP, protocolul TCP/IP folosește ARP pentru a găsi adresa fizică a destinației.

Când pachetul trebuie trimis la o adresă IP din afara rețelei, acesta este trimis către poarta de acces care este prezentă fizic în rețeaua dumneavoastră și care, prin urmare, poate răspunde la o solicitare ARP pentru o adresă fizică.

Cabluri Ethernet

Standardul original Ethernet folosește un cablu coaxial gros, cu un diametru de aproape jumătate de inch. Acesta este denumit *cablu gros (thickwire)* sau *Ethernet gros*, cu toate că standardul 802.3 al IEEE îl denumește 10BASE5. Această denumire are semnificația că rata de transmitere a datelor este de 10 megabiți pe secundă (10 Mbps), că transmisia se face în banda de bază (care înseamnă că capacitatea de purtător de semnal a cablului este dedicată numai transmișiei pachetelor Ethernet) și că lungimea totală a cablului nu poate depăși 500 de metri. Cablul gros era scump și cam masiv.

În prezent, alte două forme de cabluri Ethernet sunt mai populare. Prima alternativă la cablul Ethernet gros este *cablul subțire (thinwire)*, sau 10BASE2, care folosește un cablu coaxial flexibil subțire. Un segment de cablu Ethernet subțire poate avea maximum 185 de metri lungime. A doua alternativă, mai recentă, este cablul torsadat neecranat (UTP), cunoscut ca 10BASET.

Pentru a conecta un cablu Ethernet 10BASET, aveți nevoie de un concentrator (hub) Ethernet – o cutie hardware cu mufe RJ-45. Construiți rețeaua conectând cablul torsadat de la fiecare placă Ethernet a calculatoarelor prin acest concentrator.

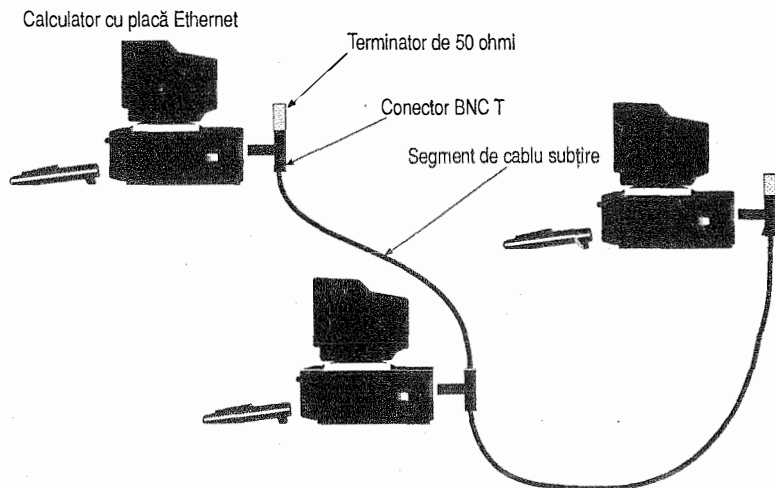


Sugestie

Cablul subțire (thinwire) are un avantaj care îl face atractiv pentru birourile mici sau pentru domiciliu în cazul în care există mai mult de un calculator. Puteți conecta în cascadă cablul subțire de la un calculator la altul și să construiți o mică rețea locală Ethernet, ca în figura 16-6.

Figura 16-6

Construirea unei mici rețele locale Ethernet cu cablu subțire



Cum se observă din figura 16-6, trebuie să dispuneți de calculatoare echipate cu plăci Ethernet. Plăcile trebuie să aibă conectori pentru cablu subțire, denumiți conectori BNC. Trebuie să aveți

de asemenea segmente de cablu subțire (cunoscute în tehnică sub numele de cablu coaxial subțire RG-58 cu impedanță de 50 de ohmi). Pentru fiecare conector BNC de placă Ethernet aveți nevoie de un conector BNC T (denumit astfel deoarece arată ca litera T) și de doi terminatori de 50 de ohmi pentru cele două capete ale rețelei Ethernet. Nu vă rămâne decât să vă construiți propria rețea, respectând conexiunile ca în figura 16-6. Eu însumi am folosit această metodă pentru a conecta câteva PC-uri și o stație de lucru în biroul meu de acasă.



Atenție!

Rețineți că trebuie să folosiți conectori BNC T și terminatori de 50 de ohmi chiar dacă conectați numai două PC-uri într-o rețea locală; nu puteți, pur și simplu, să conectați cele două plăci Ethernet cu un cablu.

Plăci Ethernet compatibile Linux

Pentru a crea o rețea locală Ethernet, aveți nevoie de o placă Ethernet pentru fiecare PC. Linux poate lucra cu o largă varietate de plăci Ethernet pentru PC-uri. În continuare, este o listă cu plăci Ethernet compatibile nucleului Linux 1.2.8.:

- 3Com 3C501 (învechită și foarte lentă)
- 3Com 3C503, 3C505, 3C507, 3C509/3C509B (ISA), 3C579 (EISA)
- 3Com Etherlink III Vortex Ethercards (3C590, 3C592, 3C595, 3C597) (PCI)
- 3Com Etherlink XL Boomerang Ethercards (3C900, 3C905) (PCI)
- 3Com 3C515 Fast Etherlink Ethercard (ISA)
- Allied Telesis AT1700 și LA100PCI-T
- AMD LANCE (79C960) și PCnet (AT1500, HP J2405A, NE1500, NE2100)
- Ansel Communications AC3200 (EISA)
- Apricot Xen-II (82596) onboard Ethernet
- AT&T GIS WaveLAN
- Cabletron E21xx
- Cogent EM110
- Danpex EN-9400
- DEC DE425 (EISA), DE434 (PCI), DE435 (PCI), DE450/DE500
- DEC DE450/DE500-XA, QSilver (driver Tulip)
- DEC DEPCA, EtherWORKS și EtherWORKS 3
- Fujitsu FMV-181/182/183/184
- HP PCLAN (27245 și alte serii 27xxx)
- HP PCLAN PLUS (27247B și 27252A)
- Adaptor intern IBM ThinkPad 300
- ICL EtherTeam 16i/32 (EISA)
- Intel EtherExpress și EtherExpress Pro
- KTI ET16/P-D2, ET16/P-DC ISA
- New Media Ethernet
- Novell Eagle NE1000 și NE2000

- PureData PDUC8028, PDI8023
- Racal-Interlan NI5210 (i82586 Ethernet chip; nu lucrează bine cu driverul curent)
- Racal-Interlan NI6510 (am 7990 LANCE Ethernet chip; are probleme cu mai mult de 16 MB de RAM)
- Schneider & Koch G16
- SEEQ 8005
- SMC Ultra/EtherEZ (ISA)
- Seriile SMC 9000
- SMC PCI EtherPower 10/100 (driver Tulip)
- Western Digital WD8003 (la fel ca SMC Elite) și WD8013 (SMC Elite Plus)
- Adaptor intern Zenith Z-Note
- Znyx 312 etherarray (driver Tulip)

Dacă ați modernizat nucleul Linux, noul nucleu poate lucra chiar cu mai multe plăci decât cele menționate în listă.

Dacă nu ați cumpărat încă o placă Ethernet, atunci cumpărați o placă pe 16 biți și nu una pe 8 biți. O placă pe 16 biți transferă date mai rapid și conține un buffer intern mai mare. Plăcile sunt livrate cu diferite tipuri de conectori, după cum urmează:

- Un conector DB-15 pentru Ethernet gros. (Aveți nevoie de un transceiver suplimentar pentru a conecta cablul gros (thickwire) la alte tipuri de Ethernet, ca de exemplu cu cablu subțire (thinwire) sau 10BASE-T.)
- Un conector BNC pentru cablu subțire
- Un conector RJ-45 pentru 10BASE-T. (Conectorul RJ-45 arată ca o mufă obișnuită de telefon RJ-11, dar versiunea RJ-45 are 8 pini în loc de 6.)



Sugestie

În prezent, multe plăci Ethernet sunt furnizate cu mai mult de un conector (chiar cu toți trei). Dacă aveți un birou mic la serviciu sau la domiciliu, vă recomand să cumpărați o placă Ethernet cu un conector pentru cablu subțire. Apoi puteți crea o rețea locală Ethernet cu câteva cabluri coaxiale RG58, conectori BNC T și terminatori de 50 de ohmi.

Rețelele cu cablu subțire sunt ușor de instalat, dar nu este convenabil să se conecteze mai multe PC-uri în acest fel. O problemă este aceea că orice întrerupere a cablului cauzează căderea întregii rețele.

Pentru a instala o rețea 10BASE-T, aveți nevoie de un concentrator separat cu mufe de telefon. Un fir pleacă din fiecare placă Ethernet a fiecărui calculator către concentrator, iar acesta stabilește conexiunea potrivită pentru Ethernet.

Puteți folosi plăci Ethernet cu un conector Ethernet gros DB-15 într-o rețea cu cablu subțire sau într-o rețea 10BASE-T. Aveți nevoie doar de un transceiver ce se atașează la portul DB-15 și furnizează tipul corect de conector (BNC pentru cablu subțire și RJ-45 pentru 10BASE-T).



Sugestie

O alegere mai bună este să cumpărați versiunea mai nouă „Combo” de plăci Ethernet care este livrată cu transceiver intern atât pentru cablu subțire cât și pentru 10BASE-T; există un conector BNC pentru o rețea cu cablu subțire și o mufă RJ-45 pentru o rețea 10BASE-T.

Plăci Ethernet incompatibile cu Linux

După cum v-ați dat seama probabil până acum, cele mai multe drivere din Linux există pentru că cineva, în mod voluntar, a scris codul aferent și l-a pus, gratuit, la dispoziția publicului. Bineînțeles că programatorul poate scrie un driver numai dacă vânzătorul îi furnizează fără restricții informațiile tehnice despre dispozitivul respectiv (amintiți-vă că Licența Publică GNU impune codului sursă să fie disponibil în mod gratuit). Din păcate, câțiva furnizori de componente hardware refuză să furnizeze informații tehnice dacă nu primesc asigurări privind confidențialitatea. Multe plăci video avansate intră în această categorie, și de aceea multe dintre ele sunt incompatibile cu Linux.

Câțiva furnizori de plăci Ethernet refuză, de asemenea, să furnizeze informații tehnice în mod gratuit. În prezent, Cabletron și Xircom sunt astfel de furnizori. Datorită lipsei datelor de programare, plăcile Ethernet Cabletron și Xircom sunt incompatibile cu Linux. Trebuie să evitați aceste plăci dacă aveți de gând să rulați Linux pe calculatorul dumneavoastră



Atenție!

Placa Cabletron E2100 este compatibilă cu Linux numai pentru că este proiectată având la bază cipul National Semiconductor DP8390. Cu toate că există un driver disponibil pentru placa Cabletron E2100, este posibil ca îmbunătățirile ulterioare aduse acestui driver să nu fie disponibile.

Facilități ale nucleului pentru Ethernet

Când urmați etapele de instalare Linux de pe CD-ROM-ul anexat acestei cărți, descrise în capitolul 1, trebuie să selectați un nucleu. De obicei, aceste nuclee includ deja facilități pentru câteva plăci de rețea.

După ce terminați de instalat Linux și reinițializați sistemul, trebuie să verificați mesajele de boot afișate pentru a vedea dacă placa Ethernet a fost detectată. (Dacă mesajele afișate la pornire dispar prea repede, le puteți vedea oricând doriți folosind comanda `dmesg | more`). Calculatorul meu are, de exemplu, o placă Ethernet 3Com 3C509, iar mesajele de boot conțin următoarele rânduri:

```
eth0: 3c509 at 0x210 tag 1 , BNC port, address 00 20 af e8 a2 25 , IRQ 10 .
3c509.c:1.12 6/4/97 becker@cesdis.gsfc.nasa.gov
```

Prima linie începe cu numele dispozitivului Ethernet (`eth0` înseamnă prima placă Ethernet), urmată de câteva informații de identificare a plăcii. Mesajul arată, de asemenea, adresa portului I/O (0x210) și IRQ (10).

A doua linie, care începe cu `3c509`, conține informații despre driverul Ethernet folosit. Linia se termină cu adresa de poștă electronică a lui Donald J. Becker, autorul celor mai multe drivere Ethernet din Linux.



Secret

Dacă nu vedeți nici un mesaj despre placa dumneavoastră Ethernet, va trebui să regenerați nucleul astfel încât să activați suportul sau să încărcați modulul driverului pentru placa dumneavoastră. Pentru a regenera nucleul, urmați etapele din capitolul 2. Când introduceți comanda `make config` pentru a configura nucleul, programul de configurare pune câteva întrebări (la care răspundeți cu Yes sau No) despre dispozitivele pentru care se dorește suport în nucleu. Deoarece multe dispozitive intră în conflict unele cu altele, ar trebui să specificați numai acele dispozitive care sunt instalate pe calculatorul dumneavoastră.

Când mi-am configurat nucleul pe calculatorul meu cu o placă Ethernet 3Com 3C509, am răspuns la întrebări în felul următor:

```

3COM cards (CONFIG_NET_VENDOR_3COM) [Y/n/?] y
3c501 support (CONFIG_EL1) [M/n/y/?] n
3c503 support (CONFIG_EL2) [M/n/y/?] n
3c509/3c579 support (CONFIG_EL3) [M/n/y/?] y
3c590/3c900 series (592/595/597/900/905) „Vortex/Boomerang” support
(CONFIG_VORTEX) [M/n/y/?] n
AMS LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [Y/n/?] n
Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [Y/n/?] n
Other ISA cards (CONFIG_NET_ISA) [Y/n/?] n
EISA, VLB, PCI and on board controllers (CONFIG_NET_EISA) [Y/n/?] n
Pocket and portable adaptors (CONFIG_NET_POCKET) [Y/n/?] n
Token Ring driver support (CONFIG_TR) [Y/n/?] n
FDDI driver support (CONFIG_FDDI) [N/y/?] n
ARCnet support (CONFIG_ARCNET) [M/n/y/?] n

```

Răspunsul prestabilit apare cu majuscule între paranteze pătrate la sfârșitul fiecărei întrebări; apăsați doar Enter pentru a accepta opțiunea prestabilită. În caz contrar, răspundeți tastând **y** or **n** (pentru a indica Yes, respectiv No). Tastați **m** pentru a genera suport sub forma unui modul de driver încărcabil.

Autoprobea Ethernet

În momentul inițializării, un nucleu cu suport Ethernet încearcă să probeze și să detecteze placa Ethernet. Examinarea implică citirea din și scrierea la anumite adrese de porturi I/O.

Cu toate că specificați o singură adresă de port I/O pentru un dispozitiv, cele mai multe dispozitive folosesc în funcționare un bloc de adrese I/O. Adresa I/O pe care o specificați este adresa de bază; restul adreselor I/O sunt porturi I/O consecutive, începând cu adresa de bază.



Secret

În funcție de numărul de adrese I/O folosite de un dispozitiv, două dispozitive riscă a se suprapune în gama de adrese I/O pe care le folosesc. Placa NE2000, de exemplu, folosește 32 de porturi I/O (adică 0x20 în notația hexazecimală). Dacă selectați 0x36 ca adresă de bază I/O pentru placa NE2000, aceasta folosește porturile de la 0x36 la 0x37F. Din păcate, portul paralel (LPT1) al PC-urilor folosește adresa de bază I/O 0x378, iar controllerul IDE secundar folosește adresele 0x376-0x376. Astfel, NE2000 configurată pentru 0x360 are acum o porțiune comună cu adresele I/O folosite de primul port paralel și de al doilea controller IDE. În acest caz, puteți împiedica apariția unor probleme configurând placa NE2000 la o adresă I/O diferită, cum ar fi 0x280.

Aceste suprapuneri de adrese de porturi I/O cauzează probleme, deoarece în timpul autoprobei nucleul poate efectua operații care nu dăunează anumitor dispozitive, dar poate determina ca altele să blocheze sistemul. De obicei, dacă placa dumneavoastră Ethernet și alte dispozitive lucrau sub DOS sau Windows, nu ar trebui să aveți probleme cu ele sub Linux (presupunând, bineînțeles, că placa dumneavoastră Ethernet poate lucra cu Linux).

Dacă sistemul dumneavoastră se blochează în timpul autoprobei, puteți exclude un anumit domeniu de adrese I/O de la testare. Pentru a exclude un domeniu de adrese în timpul autoprobei, folosiți comanda `reserve` la prompt-ul de inițializare LILO. Comanda `reserve` are următoarea sintaxă:

```
reserve=BASE-IO-PORT1,NUMPORTS1[,BASE-IO-PORT2,NUMPORTS2,...]
```

Aici, `BASE-IO-PORT1` este o adresă de port I/O în notație hexazecimală (de exemplu 0x300), iar `NUMPORTS1` este numărul de porturi (de exemplu 32) ce trebuie excluse în timpul autoprobei. Argumentele aflate între paranteze pătrate sunt opționale; ele sunt folosite pentru a specifica regiuni suplimentare ce trebuie excluse.

Dacă excludeți o gamă de adrese I/O de la autoprobară, dispozitivele de la acele adrese nu vor fi detectate. Astfel, trebuie să specificați în mod explicit parametrii aceluia dispozitiv în linia de comandă de inițializare.

Pentru plăcile Ethernet, comanda LILO ce specifică parametrii dispozitivului este `ether`, care are următorul format:

```
ether=IRQ,BASE-IO-PORT,PARAM-2,NAME
```

Tabelul 16-1 explică semnificația argumentelor. Toate argumentele sunt opționale. Driverul Ethernet preia primul argument nenumeric drept `NAME`.

Tabelul 16-1 Argumentele comenzii ether

Argument	Semnificație
IRQ	Cererea de întrerupere (IRQ) pentru placa Ethernet. Specificați un IRQ zero pentru a face driverul să autodetecteze cererea de întrerupere.
BASE-IO-PORT	Adresa portului I/O de bază pentru placa Ethernet. Specificați un BASE-IO-PORT zero pentru a face driverul să autodetecteze adresa portului I/O de bază.
PARAM-1	Semnificația depinde de driverul Ethernet. Unele drivere folosesc cei mai puțin semnificativi patru biți ai acestei valori ca nivel al mesajului de depanare. Valoarea prestabilită este 0; setați acest argument la o valoare între 1 și 7 pentru a indica gradul de explicitare al mesajului emis la depanare (7 înseamnă cel mai detaliat mesaj). Valoarea 8 oprește mesajele de depanare. Driverul AMD LANCE folosește ultimii patru biți drept canal DMA.
PARAM-2	Semnificația depinde de driverul Ethernet. Driverele 3Com 3c503 folosesc această valoare pentru a face selecția între un transceiver intern și unul extern – 0 înseamnă transceiver intern, iar 1 înseamnă că placa folosește un transceiver extern conectat la portul de Ethernet gros DB-15.
NAME	Numele driverului Ethernet (<code>eth0</code> , <code>eth1</code> etc). În mod prestabilit, nucleul folosește <code>eth0</code> ca nume al primei plăci Ethernet pe care o autoprobează. Nucleul nu este testat pentru mai mult de o placă Ethernet, pentru că acest lucru ar mări posibilitatea apariției unor conflicte cu alte dispozitive în timpul autoprobarii.

Dacă aveți o placă Ethernet la adresa I/O 0x300 și nu doriți ca ea să fie autoprobă, folosiți următoarea linie de comandă la promptul de boot LILO:

```
reserve=0x300,32 ether=0,0x300,eth0
```

Comanda `reserve` împiedică autoprobarăa a 32 de porturi I/O începând cu 0x300, în timp ce comanda `ether` specifică adresa portului de bază I/O pentru placa Ethernet. Adresele I/O exacte depind de placa Ethernet și de gama de adrese pentru care există o suprapunere cu alte dispozitive. În majoritatea cazurilor nu va trebui să folosiți aceste comenzi.



Secret

O observație despre PC Cards: adresele lor I/O și IRQ-urile sunt atribuite de BIOS-ul acestor PC Cards la pornirea sistemului. Astfel, nu puteți seta adresa I/O sau IRQ-ul pentru nici o placă PCI prin intermediul unei linii de comandă LILO. Chiar dacă specificați acești parametri prin comenzi LILO, ei sunt ignorați în cazul unui dispozitiv PCI.

Nume de dispozitive pentru rețele

Pentru cele mai multe dispozitive, Linux folosește fișiere din directorul /dev. Totuși, dispozitivele pentru rețele au nume care sunt definite intern în nucleu; pentru aceste dispozitive nu există fișiere în directorul /dev. Iată câteva nume uzuale de dispozitive pentru rețelele folosite în Linux:

- **lo** – dispozitivul de loopback. Acest dispozitiv este folosit pentru manevrarea eficientă a pachetelor de rețea trimise de sistemul dumneavoastră către el însuși (când, de exemplu, un client X comunică cu un server X din același sistem).
- **eth0** – prima placă Ethernet. Dacă aveți mai multe plăci Ethernet, ele primesc numele eth1, eth2 etc.
- **ppp0** – primul port serial configurat pentru o legătură point-to-point cu alt calculator, folosind protocolul Point-to-Point (PPP). Dacă aveți mai multe porturi seriale configurate pentru protocolul de rețea PPP, atunci acestea vor primi numele ppp1, ppp2 etc.
- **s10** – primul port serial configurat pentru Serial Line Internet Protocol (SLIP). SLIP este folosit pentru stabilirea unei legături point-to-point cu o rețea TCP/IP. Dacă folosiți al doilea port serial pentru SLIP, el primește numele de dispozitiv s11.



Referință încrucișată

Întotdeauna aveți un dispozitiv pentru loopback (lo), indiferent dacă aveți rețea sau nu. Dispozitivul pentru loopback trece datele de la un proces la altul, fără ca acestea să fie trimise către o rețea. De fapt, rolul dispozitivului de loopback este de a permite aplicațiilor de rețea să lucreze atât timp cât procesele de comunicație se desfășoară pe același sistem.

Protocolul PPP este popular în rețelele dial-up, în care folosiți un modem pentru a intra în legătură cu un calculator gazdă din Internet (de obicei un sistem de la birou sau de la furnizorul de servicii Internet), stabilind astfel conexiunea cu Internetul. Capitolul 18 tratează acest subiect în detaliu.

Dacă doriți să vedeți numele dispozitivelor de rețea instalate pe sistemul dumneavoastră, încercați următoarea comandă:

```
cat /proc/net/dev
```

Această comandă afișează numele dispozitivelor pentru rețea precum și statistici despre numărul de pachete trimise și primite de un anumit dispozitiv.

Plăci Ethernet multiple

Puteți folosi un PC Linux ca poartă de acces între două rețele Ethernet. În acest caz, puteți avea două plăci Ethernet în calculator. Nucleul Linux poate lucra cu mai mult de o placă Ethernet, dar *nu* detectează în mod automat plăcile multiple. Nucleul caută doar prima placă Ethernet. Dacă aveți două plăci Ethernet, trebuie să specificați parametrii acestora în linia de comandă de inițializare LILO. (Cele două plăci trebuie să aibă, bineînțeles, cereri de întrerupere – IRQ și adrese I/O diferite.) Iată, în continuare, o linie de comandă de boot pentru două plăci Ethernet:

```
ether=10,0x20,eth0 ether=5,0x300,eth1
```

Dacă aveți două plăci Ethernet, puteți plasa parametrii LILO de inițializare necesari în fișierul */etc/lilo.conf* pentru a nu fi necesar să introduceți argumentele de fiecare dată când porniți Linux. Pentru exemplu anterior, linia din fișierul */etc/lilo.conf* arată astfel:

```
append="ether=10 , 0x220 , eth0 ether=5 , 0 x 300 , eth1"
```

Dacă aveți de gând să folosiți un PC Linux cu două interfețe de rețea drept poartă de acces TCP/IP, trebuie să recompilați nucleul cu facilitatea IP forwarding activată.

Capitolul 2 explică cum trebuie regenerat nucleul. În timpul procesului de regenerare, când folosiți comanda `make config`, sunt afișate următoarele întrebări (acestea sunt doar câteva):

```
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?] y
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] y
IP: optimize as router not host (CONFIG_IP_ROUTER) [N/y/?] y
```

Dacă folosiți calculatorul dumneavoastră Linux drept router, trebuie să răspundeți cu **y** la aceste întrebări.

Configurarea TCP/IP în Linux

Ca la majoritatea programelor cu suport Linux, configurarea TCP/IP constă în pregătirea mai multor fișiere de configurare (fișiere text pe care le puteți edita cu orice editor de texte). Multe dintre aceste fișiere de configurare se găsesc în directorul */etc*. Programul de instalare Red Hat se face util prin ascunderea detaliilor despre fișierele de configurare TCP/IP. Cu toate acestea, este mai bine dacă cunoașteți numele fișierelor și rolul lor, astfel încât să le puteți edita manual la nevoie.



Referință încrucișată

Următoarele secțiuni prezintă modul de configurare TCP/IP pentru o rețea locală Ethernet. Capitolul 18 tratează rețelele de tip dial-up sub Linux, incluzând subiecte precum PPP.

Înainte de a examina configurarea TCP/IP, asigurați-vă că placa Ethernet a sistemului dumneavoastră este corect instalată și detectată de nucleul Linux.

Configurarea nucleului pentru TCP/IP

Configurarea nucleului este primul pas în setarea sistemului dumneavoastră pentru TCP/IP. Așa cum se arată în capitolul 2, primul pas în configurarea nucleului presupune executarea următoarelor comenzi:

```
cd /usr/src/linux
make config
```

Programul de configurare pune câteva întrebări despre diferite facilități ale sistemului; câteva din aceste întrebări se referă la rețele. În continuare, sunt prezentate primele două seturi de întrebări despre rețele (și câteva răspunsuri posibile):

```
* Opțiuni de lucru în rețea
```

```
*
```

```
Network firewalls (CONFIG_FIREWALL) [Y/n/?] n
Network aliasing (CONFIG_NET_ALIAS) [Y/n/?] n
TCP/IP networking (CONFIG_INET) [Y/n/?] y
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?] n
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] n
IP: syn cookies (CONFIG_SYN_COOKIES) [Y/n/?] y
IP: accounting (CONFIG_IP_ACCT) [Y/n/?] n
IP: optimize as router not host (CONFIG_IP_ROUTER) [N/y/?] n
IP: tunneling (CONFIG_NET_IPIP) [M/n/y/?] n
```

```
* (lasati urmatoarele optiuni nemodificate)
```

```
*
```

```
IP: PC/TCP compatibility mode (CONFIG_INET_PCTCP) [N/y/?] n
IP: Reverse ARP (CONFIG_INET_RARP) [M/n/y/?] m
IP: Disable Path MTU Discovery (normally enabled)
(CONFIG_NO_PATH_MTU_DISCOVERY) [N/y/?] n
IP: Drop source routed frames (CONFIG_IP_NOSR) [Y/n/?] y
```



```
IP: Allow large windows (not recommended if <16Mb of memory)
(CONFIG_SKB_LARGE) [Y/n/?] y
The IPX protocol (CONFIG_IPX) [M/n/y/?] n
```

Puteți accepta răspunsul prestabilit (afișat între paranteze pătrate la sfârșitul fiecărei întrebări) pentru aproape toate întrebările. Ar trebui să răspundeți cu Yes (da) pentru a activa facilitățile de IP forwarding dacă doriți să folosiți sistemul dumneavoastră Linux ca poartă de acces (sau router) între două rețele separate.

Următoarea listă explică înțelesul câtorva din aceste întrebări de configurare:

Parafocurile rețelei (Network firewalls) – Dacă răspundeți afirmativ la această întrebare, nucleul permite programelor de rețea să activeze sau să dezactiveze selectiv accesul către grupurile de porturi TCP/IP. Aveți nevoie de software suplimentar pentru a controla accesul către anumite porturi TCP/IP.

Pseudonimul rețelei (Network aliasing) – Dacă răspundeți afirmativ la această întrebare, nucleul permite multiple adrese IP pentru o singură interfață de rețea.

Protocoalele de rețea TCP/IP (TCP/IP networking) – Această opțiune permite folosirea facilităților suitei de protocoale TCP/IP în nucleu. Ar trebui să răspundeți afirmativ la această întrebare, chiar dacă nu aveți de gând să construiți o rețea locală. TCP/IP poate fi folosit chiar și în cadrul unui singur sistem; dispozitivul de loopback 10 furnizează mecanismul de transport de date pentru a sprijini schimburile TCP/IP între procesele de pe același PC.

IP: transmitere/porturi de acces (IP: forwarding/gatewaying) – Răspundeți afirmativ la această întrebare dacă doriți să folosiți sistemul dumneavoastră Linux pentru a trimite pachete TCP/IP către altă rețea. Opțiunea prestabilită este de a dezactiva transmiterea. Ar trebui să schimbați această opțiune dacă doriți să vă conectați rețeaua dumneavoastră locală la Internet prin intermediul unui calculator Linux folosind PPP.

IP: difuzare pe grupuri (IP: multicasting) – Această întrebare se referă la capacitatea sistemului de a primi și trimite pachete de rețea către un subset de calculatoare gazdă din rețea. Câteva servicii Internet noi, ca de exemplu Internet Talk Radio, se bazează pe multicasting. Chiar dacă această caracteristică este activă, aveți nevoie de software suplimentar pentru a folosi facilitățile de multicasting.

IP: syn cookies – Dacă răspundeți afirmativ la această întrebare, nucleul include un cod ce furnizează protecție contra unui tip de atac împotriva rețelei cunoscut ca inundare SYN.

IP: numărare (IP: accounting) – Dacă activați această caracteristică, nucleul numără și înregistrează volumul de date care vin și care pleacă (în baiți) pe un anumit port și pe o anumită adresă. Puteți folosi această caracteristică dacă folosiți un PC Linux pentru a vă conecta la alte sisteme prin intermediul Internetului. Puteți folosi informațiile furnizate de IP: accounting pentru a întocmi nota de plată a clienților dumneavoastră în funcție de nivelul de utilizare.

Modul de compatibilitate PC/TCP (PC/TCP compatibility mode) – Dacă sistemul dumneavoastră Linux este conectat într-o rețea locală cu alte PC-uri din care câteva rulează software de rețea PC/TCP, ar trebui să răspundeți afirmativ la această întrebare; astfel pot fi evitate unele probleme ce pot apărea atunci când un PC care rulează PC/TCP se conectează la PC-ul dumneavoastră Linux.

ARP invers (Reverse ARP) – ARP se referă la protocolul de rezoluție al adresei, care este folosit pentru a determina adresa fizică Ethernet ce corespunde unei adrese IP. Opusul lui ARP este ARP invers (Reverse ARP – RARP) – un protocol care permite unui sistem să spună „Iată adresa mea Ethernet; cineva să-mi dea o adresă IP.” Câteva stații de lucru Sun 3

și alte stații de lucru fără hard-disc au nevoie de RARP pentru a lucra. De obicei, nu trebuie să răspundeți afirmativ la această întrebare.

Protocolul IPX (The IPX protocol) – Răspundeți cu Yes dacă doriți să includeți suport pentru protocolul de schimb de pachete între rețele (Internet Packet Exchange – IPX) – care face parte din protocolul sistemului de rețea Xerox (Xerox Network System – XNS). Novell NetWare folosește IPX ca nivel de rețea. Nivelul de transport pentru NetWare se numește schimb secvențial de pachete (Sequenced Packet Exchange – SPX); Linux nu include asistență pentru SPX.

Rularea programului netcfg

După ce v-ați asigurat că nucleul Linux este corect configurat pentru TCP/IP, trebuie să vă asigurați că există fișierele de configurare corespunzătoare. Linux Red Hat (care se află pe CD-ROM-ul anexat acestei cărți) conține un program numit netcfg (în directorul /usr/bin) care vă ajută să configurați diferite interfețe de rețea pe sistemul dumneavoastră pentru a putea lucra cu TCP/IP. Puteți rula programul netcfg pentru a adăuga o nouă interfață de rețea sau pentru a modifica informații, ca de exemplu servere de nume și numele calculatoarelor gazdă (puteți, de asemenea, edita direct fișierele de configurare listate în secțiunea „Folosirea fișierelor de configurare TCP/IP”).

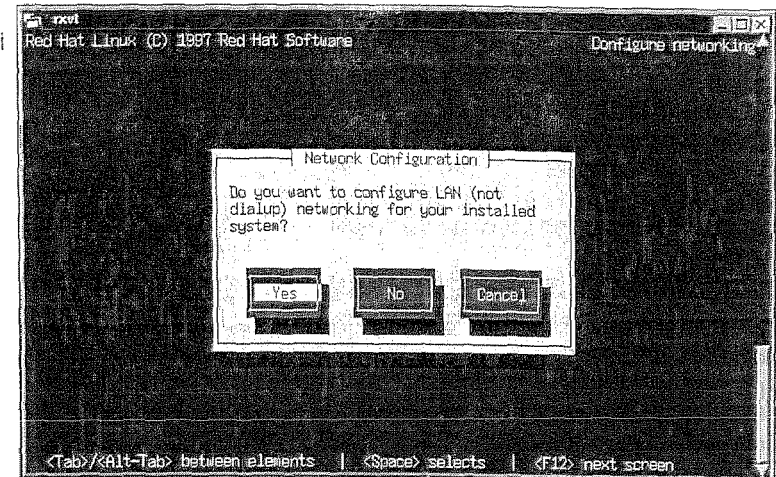
Pentru a rula netcfg, conectați-vă ca *root*, porniți X Window System și tastați următoarea comandă într-o fereastră xterm:

```
/usr/bin/netcfg
```

Programul netcfg afișează o casetă de dialog ca în figura 16-7.

Figura 16-7

Configurarea unei rețele TCP/IP cu programul netcfg



Puteți configura diferite aspecte ale rețelei cu ajutorul casetelor de dialog. Mai concret, aveți următoarele posibilități:

■ **Nume** – vă permite să introduceți numele calculatorului gazdă pentru sistemul dumneavoastră și adresele IP ale serverelor de nume. Adresele serverelor de nume sunt stocate

în fișierul `/etc/resolv.conf`. Numele calculatorului gazdă este stocat într-o variabilă ce se găsește în fișierul `/etc/sysconfig/network`.

■ **Hosts** – prezintă conținutul fișierului `/etc/hosts` și vă permite să adăugați, să ștergeți sau să editați intrări.

■ **Interfaces (Interfețe)** – vă permite să adăugați o nouă interfață de rețea, să specificați adresa IP a interfeței și să activați interfața. Această informație se depozitează în diferite fișiere din directorul `/etc/sysconfig`.

■ **Routine (Rutină)** – vă permite să adăugați rute statice (fiecare rută listează poarta de acces ce trebuie folosită pentru a ajunge la o anumită rețea).

Testarea rețelei

După ce rulați programul `netcfg`, veți dori probabil să verificați dacă rețeaua funcționează. Dacă nu ați repornit încă sistemul, trebuie să rulați programul `/sbin/ifconfig` pentru a configura interfețele Ethernet pentru adresa dumneavoastră IP. Pentru un sistem cu adresa IP 206.197.168.200, veți tasta următoarea comandă (trebuie să intrați în sistem ca `root` pentru a avea această posibilitate):

```
/sbin/ifconfig eth0 206.197.168 netmask 255.255.255.0 broadcast
206.197.168.255
```

Verificarea accesibilității altui sistem

Acum ar trebui să folosiți utilitarul `ping` pentru a verifica dacă un alt sistem din rețea este accesibil. Pe calculatorul meu, pot încerca următoarele:

```
ping 206.197.168.50
PING 206.197.168.50 (206.197.168.50): data bytes
64 bytes from 206.197.168.50: icmp_seq= 0 ttl=32 time=3.4 ms
64 bytes from 206.197.168.50: icmp_seq= 1 ttl= 32 time=1.8 ms
64 bytes from 206.197.168.50: icmp_seq= 2 ttl= 32 time=1.8 ms
64 bytes from 206.197.168.50: icmp_seq= 3 ttl= 32 time=1.9 ms
    (press Ctrl-C here)
--- 206.197.168.50 ping statistics ---
4 packets transmitted, 4 packets received, 0% packets loss
round-trip min/avg/max = 1.8/2.2/3.4 ms
```

Dacă comanda `ping` arată că alte sisteme din rețeaua dumneavoastră sunt accesibile, atunci puteți începe să utilizați alte programe de rețea, precum `ftp` și `telnet`.

Folosirea fișierelor de configurare TCP/IP

Rularea programului `netcfg` este suficientă pentru a configura TCP/IP pe sistemul dumneavoastră. Probabil că doriți să știți care sunt fișierele de configurare, astfel încât să le puteți edita în caz de nevoie. Puteți specifica serverele de nume în fișierul `netcfg`, de exemplu, dar uneori este necesar să adăugați un server de nume alternativ. Pentru aceasta, trebuie să aveți cunoștințe despre fișierul `/etc/resolv.conf`, care stochează adresele IP ale serverelor de nume.

Următoarele secțiuni descriu fișierele de configurare TCP/IP de bază.

/etc/HOSTNAME

Fișierul `/etc/HOSTNAME` stochează numele calculatorului dumneavoastră gazdă – întregul nume de domeniu pentru sistemul dumneavoastră, ca de exemplu `addlab.nws.noaa.gov`.



Secret

În Linux Red Hat, numele calculatorului gazdă este stocat într-o variabilă numită `HOSTNAME` din fișierul `/etc/sysconfig/network`. În timpul pornirii sistemului, fișierul de inițializare (`/etc/rc.d/rc.sysinit`) folosește comanda `hostname` pentru a seta numele calculatorului dumneavoastră gazdă la valoarea variabilei `HOSTNAME` din fișierul `/etc/sysconfig/network`. Fișierul de comenzi `/etc/rc.d/rc.sysinit` salvează, de asemenea, numele calculatorului gazdă în fișierul `/etc/HOSTNAME`.

/etc/hosts

Fișierul text `/etc/hosts` conține o listă de adrese IP și nume de calculatoare gazdă pentru rețeaua dumneavoastră locală. În absența unui server de nume, orice program de rețea din sistemul dumneavoastră consultă acest fișier pentru a determina adresa IP ce corespunde numelui unui calculator gazdă.

Iată, în continuare, fișierul `/etc/hosts` din sistemul meu, care arată adresele IP și numele altor calculatoare gazdă din rețeaua mea locală:

```
127.0.0.1 localhost
# Alte calculatoare gazda din retea locala
206.197.168.200 lnbsoft.com lnbsoft
206.197.168.50 lnb386
206.197.168.100 lnb486
206.197.168.150 lnbmac
206.197.168.2 lnbSun
```

Sfarsitul listei cu calculatoare gazda

Așa cum arată exemplul, fiecare linie din fișier începe cu o adresă IP, urmată de numele calculatorului gazdă corespunzător acesteia. Puteți avea mai mult de un nume de calculator gazdă pentru o adresă IP dată.

/etc/networks

Fișierul `/etc/networks` este alt fișier text ce conține numele și adresele IP ale rețelelor. Aceste nume de rețele sunt folosite de obicei în comanda de rutare (`/sbin/route`) pentru a indica o rețea după nume în loc de adresa ei IP.

Nu vă alarmați dacă PC-ul dumneavoastră Linux nu conține fișierul `/etc/networks`. Rețeaua dumneavoastră TCP/IP va lucra bine și fără acest fișier. De fapt, în Linux Red Hat programul de instalare nu creează nici un fișier `/etc/networks`.

/etc/host.conf

Linux folosește o bibliotecă de rezoluție pentru a obține adresele IP corespunzătoare numelui unui calculator gazdă. Fișierul `/etc/host.conf` specifică modul de rezoluție a numelor. Un fișier tipic `/etc/host.conf` poate conține următoarele linii:

```
order hosts, bind
multi on
```

Intrările din fișierul `/etc/host.conf` indică bibliotecii de rezoluție ce servicii să folosească și în ce ordine, pentru a stabili numele.

Opțiunea `order` indică ordinea serviciilor. Intrarea cu caracter demonstrativ specifică faptul că biblioteca de rezoluție ar trebui mai întâi să consulte fișierul `/etc/hosts` și apoi să verifice serverul de nume pentru a se decide asupra unui nume.



* Vădit

Opțiunea `multi` determină dacă un calculator gazdă din fișierul `/etc/hosts` poate avea multiple adrese IP. Calculatoarele gazdă care au mai multe adrese IP se spune că au mai multe adrese (*multihomed*), deoarece prezența adreselor multiple IP implică faptul că sistemul gazdă are câteva interfețe de rețea. (calculatorul gazdă există în mai multe rețele simultan)

/etc/resolv.conf

Fișierul `/etc/resolv.conf` este un alt fișier text folosit de către utilitarul de rezoluție – o bibliotecă ce determină adresa IP pentru un calculator gazdă. Iată un exemplu de fișier `/etc/resolv.conf`:

```
domain com
nameserver 164.109.1.3
nameserver 164.109.10.23
```

Prima linie specifică numele domeniului pentru sistemul dumneavoastră. Linia `nameserver` indică adresele IP pentru serverele de nume ale domeniului dumneavoastră. Dacă aveți servere de nume multiple, ar trebui să le scrieți pe linii separate. Ele sunt interogate în ordinea în care apar în fișier.



Observație

Dacă nu aveți nici un server de nume pentru rețeaua dumneavoastră, puteți ignora acest fișier. TCP/IP ar trebui să funcționeze, chiar dacă nu puteți face referire la calculatorul gazdă după nume.

/etc/hosts.allow

Acest fișier specifică calculatoarele gazdă cărora li se permite să folosească serviciile Internet (ca de exemplu Telnet și FTP) care rulează pe sistemul dumneavoastră. Așa cum este explicat în secțiunea „Superserverul `inetd`”, programul `/usr/sbin/tcpd` consultă fișierul `/etc/hosts.allow` înainte de a porni serviciile Internet. Programul pornește serviciile numai dacă intrările în fișierul `hosts.allow` indică faptul că sistemul gazdă care le solicită are permisiune de utilizare pentru aceste servicii.

Intrările în fișierul `/etc/hosts.allow` sunt de forma *server:adresa IP*, unde *server* se referă la numele programului ce furnizează un anumit serviciu Internet, iar *adresa IP* identifică calculatorul gazdă care are permisiunea de a folosi aceste servicii. De exemplu, dacă doriți ca toate calculatoarele gazdă din rețeaua dumneavoastră locală (care are adresa de clasă C 192.168.1.0) să acceseze serviciul FTP (care este furnizat de programul `in.ftpd`), trebuie să adăugați următoarea linie în fișierul `/etc/hosts.allow`:

```
in.ftpd:192.168.1.
```

Dacă doriți ca toate calculatoarele gazdă din rețeaua locală să aibă acces la toate serviciile Internet, puteți folosi cuvântul cheie `ALL` și puteți rescrie linia astfel:

```
ALL:192.168.1
```

Pentru a face accesibile toate serviciile Internet către toate calculatoarele gazdă, puteți înlocui adresa IP cu `ALL`, astfel:

```
ALL:ALL
```

Puteți, de asemenea, să folosiți nume de calculatoare gazdă în locul adreselor IP.



Sugestie

Pentru a învăța sintaxa detaliată a intrărilor din fișierul `/etc/hosts.allow`, tastați la promptul interpretorului din Linux comanda `man hosts_access`.

/etc/hosts.deny

Acest fișier este exact opusul fișierului `/etc/hosts.allow` – în timp ce `hosts.allow` specifică ce calculator gazdă poate accesa serviciile Internet (ca de exemplu Telnet și FTP) din sistemul dumneavoastră, fișierul `hosts.deny` identifică calculatoarele gazdă cărora le sunt interzise aceste servicii. Așa cum este explicat în secțiunea „Superserverul `inetd`”, programul `/usr/sbin/tcpd` consultă fișierul `/etc/hosts.deny` în cazul în care nu găsește vreo regulă în fișierul `/etc/hosts.allow` referitoare la calculatorul gazdă care a făcut solicitarea. Programul `tcpd` interzice folosirea serviciilor dacă găsește în fișierul `hosts.deny` o regulă referitoare la calculatorul gazdă.

Intrările din fișierul `/etc/hosts.deny` au același format ca cele din fișierul `/etc/hosts.allow` – ele sunt de forma *server:adresa IP*, unde *server* se referă la numele programului ce furnizează un anumit serviciu Internet, iar *adresa IP* identifică calculatorul gazdă ce are permisiunea de a folosi aceste servicii.

Dacă deja ați setat intrările în fișierul `/etc/hosts.allow` pentru a permite accesul anumitor calculatoare gazdă, puteți plasa următoarea linie în fișierul `/etc/hosts.deny` pentru a interzice celorlalte calculatoare gazdă accesul la orice serviciu din sistemul dumneavoastră:

```
ALL:ALL
```



Sugestie

Pentru a învăța sintaxa detaliată a intrărilor în fișierul `/etc/hosts.deny`, tastați la prompt-ul interpretorului din Linux comanda `man hosts_access`.

Configurarea rețelelor în momentul inițializării

Doriți să porniți rețeaua automat de fiecare dată când inițializați sistemul. Pentru a realiza acest lucru, trebuie să introduceți anumite comenzi în unul sau mai multe fișiere script folosite la inițializare. Procesul `init` rulează imediat după încărcarea sistemului de operare Linux. Procesul consultă fișierul `/etc/inittab` și apoi execută diferite comenzi (de obicei fișiere de comenzi), în funcție de nivelul curent de rulare. În nivelul de rulare 3 – nivelul multiutilizator – `/etc/inittab` specifică faptul că `init` ar trebui să ruleze fișierul de comenzi `/etc/rc.d/rc` cu argumentul 3.

În esență, fișierul script de inițializare execută fișierul script `/etc/rc.d/init.d/network` pentru a activa toate interfețele de rețea. Dacă veți consulta fișierul `/etc/rc.d/init.d/network`, veți observa că inițializarea rețelei se execută folosind alt set de fișiere din directorul `/etc/sysconfig`. Fișierul script ce activează rețeaua verifică variabilele definite în fișierul `/etc/sysconfig/network` pentru a se decide dacă să activeze rețeaua. În fișierul `/etc/sysconfig/network` ar trebui să vedeți o linie ce conține variabila `NETWORKING`, astfel: `NETWORKING=yes`

Rețeaua este activată numai dacă variabila `NETWORKING` are valoarea `yes`.

Fișierul script `/etc/rc.d/init.d/network`, la rândul lui, execută un număr de fișiere script din directorul `/etc/sysconfig/network-scripts` pentru a activa anumite interfețe de rețea. De exemplu, pentru a activa interfața Ethernet `eth0`, script-ul `/etc/sysconfig/network-scripts/ifup` este executat cu `/etc/sysconfig/network-scripts/ifcfg-eth0` ca fișier de configurare. Iată ce conține un fișier tipic `/etc/sysconfig/network-scripts/ifcfg-eth0`:

```
DEVICE=eth0
```

```
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=none
```

După cum puteți vedea, acest fișier conține numele dispozitivului de rețea, adresa IP a interfeței și câțiva parametri TCP/IP. Variabila ONBOOT indică dacă această interfață de rețea trebuie activată la încărcarea sistemului de operare Linux. În cazul plăcilor Ethernet, ați dori să activați interfața în momentul inițializării; prin urmare, variabila ONBOOT trebuie setată pe *yes*.



Referință
încrucișată

Fișierele din directorul `/etc/sysconfig` sunt create de programul de instalare Linux Red Hat dacă urmați etapele prezentate în capitolul 1.

Fișierul script `/etc/sysconfig/network-scripts/ifup` execută, în primul rând, următoarele comenzi:

- `/sbin/ifconfig`, pentru a configura interfața de rețea specificată; în acest caz, placa Ethernet (`eth0`).
- `/sbin/route`, pentru a seta tabelul de rute pentru interfața rețelei activate.

Diagnosticarea TCP/IP

După ce configurați nucleul pentru Ethernet și TCP/IP și după ce rulați programul `netcfg` pentru a seta rețeaua TCP/IP, puteți folosi diferite aplicații de rețea fără probleme. Suita de protocoale TCP/IP include câteva instrumente care vă ajută să monitorizați și să diagnosticați problemele.

Verificarea interfețelor

Folosiți comanda `/sbin/ifconfig` pentru a vedea cum este configurată interfața de rețea curentă. Comanda `ifconfig` este folosită pentru a configura o interfață de rețea (adică să asocieze o adresă IP cu un dispozitiv de rețea). Dacă executați comanda `ifconfig` fără alte argumente în linia de comandă, este afișată informația despre interfața de rețea curentă. Următoarele rânduri prezintă ceea ce se obține prin apelarea comenzii `ifconfig`:

```
/sbin/ifconfig
lo Link encap: Local Loopback
  inet addr: 127.0.0.1 Bcast: 127.255.255.255 Mask: 255.0.0.0
  UP BROADCAST LOOPBACK RUNNING MTU:3584 Metric:1
  RX packets: 1584 errors:0 dropped:0 overruns:0
  TX packets: 1584 errors:0 dropped:0 overruns:0

eth0 Link encap: Ethernet HWaddr 00:20:AF:E8:A2:25
  inet addr: 192.168.1.1 Bcast: 192.168.1.255
  Mask: 255.255.255.0
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets: 129187 errors:6 dropped:6 overruns:0
  TX packets: 105955 errors:0 dropped:0 overruns:0
  Interrupt: 10 Base address: 0x210
```

Rândurile de mai sus arată că două interfețe – interfața de loopback (`lo`) și o placă Ethernet (`eth0`) – sunt active pe acest sistem. Pentru fiecare interfață, puteți vedea adresa IP precum și statistici despre pachetele trimise și primite. Pentru placa Ethernet, `ifconfig` afișează, de asemenea, IRQ (`lo`) și adresa portului de bază I/O (`0x210`).

Verificarea tabelii de rutare IP

Cealaltă comandă de configurare a rețelei, `/sbin/route`, furnizează de asemenea informații de stare când este rulată fără alte argumente în linia de comandă. Dacă aveți necazuri atunci când verificați o legătură cu alt calculator gazdă (pe care îl specificați prin intermediul unei adrese IP), verificați tabela de rutare IP pentru a vedea dacă este specificată o poartă de acces prestabilită. Apoi verificați tabela de rutare a porții de acces, pentru a vă asigura de existența în acest tabel a unor căi către o rețea exterioară.

Mesajul obișnuit emis la invocarea comenzii `/sbin/route` arată astfel:

```
/sbin/route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default-gateway * 255.255.255.255 UH 0 0 0 ppp0
192.168.1.0 * 255.255.255.0 U 0 0 99 eth0
127.0.0.0 * 255.0.0.0 U 0 0 54 lo
default default-gateway 0.0.0.0 UG 0 0 52 ppp0
```

Conform acestei tabeli de rutare, rețeaua locală folosește interfața Ethernet `eth0`, iar poarta de acces prestabilită este o conexiune PPP (numită `ppp0`) la Internet.

Verificarea conectivității la un calculator gazdă

Pentru a verifica o cale a rețelei către un anumit calculator gazdă, folosiți comanda `ping`. Aceasta este un instrument TCP/IP foarte folosit ce utilizează o serie de mesaje ICMP (Internet Control Message Protocol). (ICMP emite un mesaj ecou la care răspunde fiecare calculator gazdă.) Folosind mesajele și răspunsurile ICMP, `ping` poate determina dacă celălalt sistem este activ și poate calcula întârzierea dus-întors în comunicația cu acel sistem.

Următorul exemplu prezintă modul de rulare a comenzii `ping` pentru a verifica dacă unul din sistemele din rețeaua mea este activ:

```
ping 206.197.168.50
PING 206.197.168.50 (206.197.168.50): 56 data bytes
64 bytes from 206.197.168.50: icmp_seq= 0 ttl=32 time=3.4 ms
64 bytes from 206.197.168.50: icmp_seq= 1 ttl= 32 time=1.8 ms
64 bytes from 206.197.168.50: icmp_seq= 2 ttl= 32 time=1.8 ms
64 bytes from 206.197.168.50: icmp_seq= 3 ttl= 32 time=1.9 ms
  (press Ctrl-C here)
--- 206.197.168.50 ping statistics ---
4 packets transmitted, 4 packets received, 0% packets loss
round-trip min/avg/max = 1.8/2.2/3.4 ms
```

Pe unele sisteme, `ping` pur și simplu raportează că un calculator gazdă este activ. Mai puteți obține informații despre durata comunicării între sisteme folosind argumentele adecvate în linia de comandă. În Linux, comanda `ping` continuă să ruleze până când tastați `Ctrl-C` pentru a o opri; apoi afișează o scurtă statistică, ce prezintă timpul tipic necesar transmiterii unui pachet între două sisteme.

Verificarea stării rețelei

Pentru a verifica starea rețelei folosiți comanda `netstat`. Această comandă afișează starea diferitelor tipuri de conexiuni ale rețelei (ca de exemplu TCP sau UDP). Puteți vedea imediat starea interfețelor folosind opțiunea `-i`, astfel:

```
netstat -i
Kernel Interface table
```

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flags
Lo	3584	0	1588	0	0	0	1588	0	0	0	BLRU
eth0	1500	0	129287	6	6	0	106008	0	0	0	BRU

În acest caz, mesajul emis arată starea curentă a interfețelor de loopback și Ethernet. Tabelul 16-2 descrie semnificația fiecărei coloane.

Tabelul 16-2 Coloane în tabelul interfeței nucleului

Coloana	Semnificație
Iface	Numele interfeței
MTU	Unitatea maximă de transfer – numărul maxim de baiți pe care-l poate conține un pachet
RX-OK, TX-OK	Numărul pachetelor fără erori primite (RX) sau transmise (TX)
RX-ERR, TX-ERR	Numărul pachetelor cu erori
RX-DRP, TX-DRP	Numărul pachetelor ignorate
RX-OVR, TX-OVR	Numărul pachetelor pierdute datorită fluxului prea mare
Indicatori	A = primește difuzare multiplă, B = transmiterea permisă, D = depanarea activată, L = interfața loopback (observați indicatorul pe lo), M = toate pachetele primite, N = caracterele trailer evitate, O = nici un ARP pe această interfață, P = interfață point-to-point, R = interfața rulează, U = interfața este activă.

Altă opțiune importantă a comenzii netstat este -t, care arată toate conexiunile TCP active. Următoarele rânduri sunt rezultatul invocării comenzii netstat -t pe unul din calculatoarele mele Linux:

```
netstat -t
Active Internet connections ( w/o servers )
Proto Recv-Q Send-Q Local Address   Foreign Address State
tcp        0      0 lnbp75:telnet   lnbp133:listen  ESTABLISHED
tcp        0    124 lnbp75:telnet   lnbp200:listen  ESTABLISHED
tcp        0      0 lnbp75:ftp      lnbp486:listen  ESTABLISHED
```

În acest caz, coloanele afișate arată protocolul (Proto), numărul baiților din coada de primire și transmitere (Recv-Q, Send-Q), portul local TCP în formatul hostname:service (adresa locală), portul de la distanță (adresa străină) și starea conexiunii.

Rezumat

Linux include suport multiplu pentru rețelele TCP/IP și Ethernet. Rețelele Ethernet cu cablu subțire (thinwire Ethernet) care folosesc cabluri coaxiale RG-58 oferă un mod convenabil de construire a unei mici rețele, pentru că permite conectarea în cascadă a calculatoarelor. Acest capitol explică noțiunile de bază despre TCP/IP și Ethernet; de asemenea, prezintă modul de configurare a unei rețele TCP/IP pe calculatorul dumneavoastră Linux. Din lectura acestui capitol ați învățat următoarele:

- Modelul pe 7 niveluri OSI oferă un cadru care permite diferitelor rețele să poată lucra împreună. Modelul pe niveluri OSI stabilește, de asemenea, platforma pentru diferite protocoale de rețea.
- Protocolul de control al transmisiei (Transmission Control Protocol) și protocolul Internet (TCP/IP) care își au originile în cercetările inițiate de Agenția de Proiecte de Cercetare

Avansată (ARPA – Advanced Research Project Agency) a Guvernului SUA în anii '70. Internetul modern a evoluat pornind de la tehnologia de rețea dezvoltată în acel timp.

- Documentația despre toate protocoalele Internet se găsește în documentele RFC (Request for Comments). Documentele RFC sunt disponibile pe Internet la adresa <ftp://rs.internic.net/rfc>. Toate standardele se găsesc în documente RFC, dar multe dintre acestea furnizează, pur și simplu, informații destinate comunității Internet.
- Comunicarea între rețele este baza protocolului TCP/IP; acesta este scopul protocolului Internet (IP). Protocolul TCP/IP identifică un calculator gazdă cu o adresă IP pe 32 de biți formată din două părți: adresa unei rețele și adresa unui calculator gazdă.
- O adresă IP, în mod obișnuit, este exprimată într-o notație zecimală cu punct, în care valoarea fiecărui bait este scrisă în formatul zecimal și separată de baitul adiacent printr-un punct (.). O adresă IP tipică este 140.90.23.100.
- Adresele IP sunt grupate în clase. Adresele din clasa A folosesc un bait pentru adresa rețelei și 3 baiți pentru adresa calculatorului gazdă. Adresele din clasa B folosesc câte 2 baiți pentru adresele de rețea și de calculator gazdă; adresele din clasa C folosesc 3 baiți pentru adresa rețelei și un singur bait pentru adresa calculatorului gazdă. Valorile primului bait indică clasa adresei: 1-126 clasa A, 128-191 clasa B și 192-223 clasa C.
- Adresele IP se epuizează rapid. Pentru a rezolva această problemă, organizația Internet Engineering Task Force a adoptat o nouă schemă de adresare pe 16 baiți (128 de biți) cunoscută ca IPv6 (sau IP versiunea 6). Calculatoarele gazdă ce folosesc adrese conform schemei IPv6 pot lucra cu cele care folosesc adresele mai vechi IPv4 (32 de biți).
- Ethernet este un cunoscut mecanism fizic pentru transportul datelor. Există câteva standarde Ethernet, fiecare din acestea folosind un tip diferit de cablu. Ethernet 10BASE5 (Ethernetul original) folosește cablu coaxial gros, 10BASE2 folosește cablu coaxial subțire iar 10BASE-T folosește cablu torsadat necranat (UTP) sau cablu de telefon. Ethernetul cu cablu subțire (10BASE2) este ușor de implementat și convenabil pentru rețele mici.
- Pentru a vă putea conecta la o rețea Internet, aveți nevoie de o placă Ethernet pe PC-ul dumneavoastră. Linux poate opera cu o mare diversitate de plăci Ethernet. Pentru a obține performanțe bune ar trebui să vă procurați o placă Ethernet pe 16 biți.
- Configurarea TCP/IP în Linux necesită setarea diferitelor fișiere de configurare. Programul *netcfg* oferă o modalitate convenabilă pentru setarea acestor fișiere. Pentru a seta TCP/IP pe sistemul dumneavoastră, aveți nevoie de câteva informații – ca de exemplu o adresă IP, adresa unei porți de acces și adresa unui server de nume. Dacă nu aveți de gând să vă conectați rețeaua locală la Internet, puteți folosi o gamă de adrese IP (ca de exemplu de la 192.168.0.0 la 192.168.255.255), fără să fie necesar să vă coordonați cu o altă organizație.
- Linux are multe facilități TCP/IP, ca de exemplu ftp (File Transfer Protocol – protocol de transfer al fișierelor) și telnet (pentru conectarea la alt sistem din rețea).
- Pentru a diagnostica problemele rețelelor TCP/IP, puteți folosi comenzile ping, route și netstat.

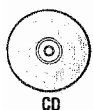
Capitolul 17

PC Cards

În acest capitol

- ▶ Veți afla generalități despre PCMCIA
- ▶ Veți înțelege noțiunea de PC Cards
- ▶ Veți examina utilizările specifice PC Cards
- ▶ Veți activa serviciile pentru plăcile PCMCIA (PCMCIA Card Services) în Linux
- ▶ Veți examina tipurile de PC Cards compatibile PCMCIA Card Services
- ▶ Veți obține cele mai recente informații despre Card Services în Linux

PCMCIA este prescurtarea de la *Personal Computer Memory Card International Association* (Asociația Internațională a Producătorilor de Plăci de Memorie pentru PC-uri), o organizație care a standardizat interfața pentru adăugarea de plăci de memorie la laptop-uri. Cu toate că inițial au fost concepute ca plăci de memorie, dispozitivele PCMCIA sunt frecvent întâlnite ca diverse componente add-on pentru laptop-uri. În prezent, laptop-urile folosesc dispozitive PCMCIA precum modemuri, plăci de rețea, controllere SCSI și plăci de sunet. Utilizarea Linux pe un laptop înseamnă lucrul cu dispozitive PCMCIA sau PC Cards, așa cum sunt ele denumite în prezent în literatura de specialitate. Datorită eforturilor lui David Hinds, puteți folosi dispozitive PCMCIA sub Linux folosind PCMCIA Card Services din Linux. Acest software este disponibil acum și este folosit de mulți oameni. Capitolul de față descrie pe scurt acest pachet de servicii PCMCIA pentru Linux.



CD

Linux Red Hat de pe CD-ROM-ul anexat acestei cărți include suport pentru dispozitivele PCMCIA. Programele de software pentru PCMCIA se prezintă sub forma unor module de driver încărcabile. Pentru a obține cea mai recentă versiune, va trebui să descărcați programul de la adresa

<ftp://sunsite.unc.edu/pub/Linux/kernel/pcmcia/>.



Observație

În continuare, mă voi referi la plăcile PCMCIA denumindu-le PC Cards, pentru că este numele corect pentru aceste dispozitive. PCMCIA se referă la organizația care specifică standardul pentru PC Cards. Totuși, voi folosi termenul de plăci PCMCIA atunci când mă voi referi la *PCMCIA Card Services for Linux* (sau, pe scurt, *Card Services*), care reprezintă programul software care suportă PC Cards în Linux.

Generalități despre PC Cards

PC Cards au fost la origine plăci de memorie statică cu acces aleator (SRAM) și plăci de memorie flash cu acces aleator (flash random access memory – flash RAM) care erau folosite pentru a stoca datele pe laptop-uri mici. Plăcile de mărimea unei cărți de credit intră într-un slot situat pe una din laturile laptop-ului. Plăcile de memorie flash foloseau memorie numai pentru citire cu ștergere și programare electrică (EEPROM), pentru a furniza laptop-urilor capacitate de stocare ce ar fi fost prea mică pentru alte medii convenționale.

Producătorii și-au dat seama imediat de avantajele slot-ului pentru placa de memorie folosit ca slot de extensie pentru laptop-uri. PCMCIA a standardizat câteva caracteristici ale PC Cards, inclusiv interfața electrică, dimensiunile plăcii și ale slotului. Această standardizare a contribuit la răspândirea plăcilor PC Cards pe piața laptop-urilor.

Până acum, sloturile PCMCIA sunt o caracteristică comună pentru aproape toate laptop-urile. Placa de memorie reprezintă doar o mică parte din întreaga piață a plăcilor PC Cards. Cele mai multe laptop-uri au sloturi PCMCIA pentru a permite adăugarea de componente hardware, precum fax/modemuri, plăci de sunet, plăci de rețea și chiar hard-discuri.



Pentru a învăța mai multe despre PCMCIA (ca organizație) și PC Cards, căutați folosind browserul dumneavoastră favorit World Wide Web la adresa <http://www.pc-card.com/>.

Specificații fizice pentru PC Cards

PC Cards sunt clasificate în trei categorii distincte, după grosimea plăcii. În continuare, sunt prezentate dimensiunile fizice pentru fiecare tip de placă PC în formatul lățime x lungime x grosime.

- Tipul I : 54 mm x 85,6 mm x 3,3 mm
- Tipul II: 54 mm x 85,6 mm x 5 mm
- Tipul III: 54 mm x 85,6 mm x 10,5 mm

Toate cele 3 tipuri de PC Cards au aceeași lungime și lățime, caracteristice unei cărți de credit, cu excepția colțurilor rotunjite; tipurile de plăci sunt diferențiate prin grosime.



Observație

Termenul de *factor de formă* este deseori folosit referitor la dimensiunile plăcilor PC Cards.

Toate PC Cards folosesc același conector pe 68 de pini. Datorită acestui conector, o placă mai subțire (de tipul I de exemplu) poate fi folosită într-un slot mai gros (de tipul II). După cum probabil v-ați dat seama, o placă mai groasă nu poate fi folosită într-un slot mai subțire.

Folosirea plăcilor PC Cards

Fiecare model de placă PC este folosit pentru un anumit tip de aplicație. Iată aplicațiile tipice ale plăcilor PC, în funcție de tipul plăcii:

- *PC Cards de tipul I.* Aceste plăci subțiri sunt folosite pentru dispozitive de memorie ca de exemplu RAM static sau RAM flash.
- *PC Cards de tipul II.* Aceste plăci sunt folosite pentru dispozitive de intrare și ieșire, ca de exemplu fax/modemuri, adaptoare de rețea și plăci de sunet.
- *PC Cards de tipul III.* Aceste plăci sunt folosite pentru dispozitive care necesită grosime mai mare, ca de exemplu hard-discuri care au componente aflate în rotație (greu de crezut, nu-i așa?).

O placă PC Card poate avea o lungime maximă de 135,6 mm (adică puțin peste 5,25 inch), ceea ce înseamnă că placa se poate extinde în afara calculatorului gazdă. Plăcile extinse sunt folosite în dispozitive precum suporturi amovibile de memorie, transceivere și antene.

Standarde PCMCIA

Toate aceste specificații sunt descrise în Standardul PCMCIA, din care au apărut până acum trei versiuni mai importante:

- *Standardul PCMCIA versiunea 1.0 (iunie 1990)*. Standardul inițial a definit conectorul cu 68 de pini și PC Cards de tipul I și II. Standardul a definit, de asemenea, structura de informații despre plăci (Card Information Structure – CIS) care a reprezentat baza interoperabilității plăcilor PC Cards. Prima versiune a standardului PCMCIA nu a luat în considerare plăcile I/O, ci numai plăcile de memorie.
- *Standardul PCMCIA versiunea 2.0, 2.01, 2.1 (1991-1994)*. A doua versiune a standardului a definit o interfață I/O pentru conectorul cu 68 de pini. Versiunea 2.01 a adăugat specificația de atașare a plăcilor PC Cards (PC Card AT Attachment – ATA) și a furnizat o versiune inițială a specificației pentru servicii pentru plăci și socluri (Card and Soclu Services – CSS). Versiunea 2.1 a continuat să îmbunătățească specificația CSS.
- *Standardul pentru PC Cards (februarie 1995)*. Această versiune a standardului anulează vechea denumire de standard PCMCIA și folosește denumirea de standard pentru PC Cards. El adaugă informații pentru a îmbunătăți compatibilitatea între diferite tipuri de PC Cards și include suport pentru facilități precum operații pe 3,3 volți, acces direct la memorie (direct memory access – DMA) și controlul magistralelor pe 32 de biți CardBus.

Terminologia plăcilor PC Cards

Deoarece toți producătorii de laptop-uri au adoptat sloturile pentru PC Cards, piața plăcilor pentru PC-uri a înregistrat o creștere explozivă. Datorită standardelor PCMCIA, plăcile PC Cards pot fi folosite în orice slot de placă PC Card. Folosind PC Cards veți întâlni câțiva termeni ce au o semnificație specială:

- *Card Information Structure (CIS)* descrie caracteristicile și facilitățile unei PC Card, astfel încât sistemul de operare sau driverul de software să poată configura placa.
- *CardBus* este o specificație electrică care descrie folosirea tehnologiei de control al magistralei și permite plăcilor PC Cards să poată lucra la o frecvență de până la 33 MHz.
- *Direct Memory Access (DMA)* are aceeași semnificație ca și la alte periferice, dar acum plăcile PC Cards pot folosi tehnologia DMA.
- *Execute In Place (XIP)* se referă la caracteristica ce permite sistemului de operare și aplicației de software să ruleze direct de pe placa PC Card, fără a trebui să fie încărcată în memoria RAM a sistemului (eliminând astfel necesitatea unui volum mai mare de RAM).
- *Low Voltage Operation* se referă la capacitatea plăcilor PC Cards de a lucra la 3,3 volți (precum și la 5 volți). Conectorul are o cheie fizică prin care se asigură că utilizatorul nu poate introduce din greșeală o placă de 3,3 volți într-un slot de 5 volți.
- *Multifunction Capability* permite unei plăci PC Card să opereze cu mai multe funcții. De exemplu, 3Com 3C562 este o placă Ethernet 10BASE-T și un modem cu rata de transfer de 28.800 bps, într-o placă PC Card cu factorul de formă de tip II.
- *Plug and Play* vă permite să introduceți sau să extrageți o placă PC Card în timp ce sistemul este pornit (acest procedeu este cunoscut ca *hot swapping* – schimbare la cald). Acest schimb la cald al plăcilor PC Cards se realizează construind pini de conectare la alimentare de lungimea cea mai mare, astfel încât liniile de date se deconectează înaintea alimentării.

■ *Power Management* se referă la capacitatea plăcilor PC Card de a interfața facilitățile de gestionare avansată a consumului de energie (APM) ale laptop-urilor, prin intermediul specificației Card Service Specification.

■ *Zoomed Video (ZV)* se referă la o conexiune între o placă PC Card și controllerul video al sistemului ce permite plăcii să scrie date video direct la controllerul video.

Programul PCMCIA Card Services în Linux

Standardizarea plăcilor PC Card semnifică faptul că dezvoltatorii sistemului Linux au putut intra în posesia informațiilor de programare necesare pentru a scrie driverele pentru plăcile PC Card. În particular, specificația Card Services Specification furnizează o interfață de programare a aplicațiilor (Application Programming Interface – API), independentă de componenta hardware ce controlează soclurile plăcilor PC Card.

O specificație corelată este Socket Services Specification, care, de asemenea, furnizează o interfață de programare a aplicațiilor pentru accesarea componentelor hardware ce controlează soclurile pentru PC Cards.

Nu trebuie să învățați despre API-urile pentru PC Cards și pentru Socket Services. David Hinds s-a ocupat deja de acest lucru, în cadrul programului PCMCIA Card Services for Linux, un pachet software pe care-l puteți folosi pentru a accesa dispozitivele PC Cards în Linux. Tot ceea ce trebuie să faceți este să activați suportul PCMCIA atunci când aveți nevoie de el.

Activarea programului Card Services

Pachetul software PCMCIA Card Services ar trebui să fie deja instalat pe sistemul dumneavoastră, deoarece Linux Red Hat include suport PCMCIA. În Linux Red Hat activarea acestui program este controlată de următoarele fișiere:

■ `/etc/rc.d/init.d/pcmcia` este script-ul de interpretor folosit pentru a lansa programul. În primul rând, comanda `/etc/rc.d/init.d/pcmcia start` este folosită pentru a activa programul la încărcarea sistemului de operare. Script-ul încarcă modulele adecvate ale driverului PCMCIA folosind comanda `sbin/insmod`. Apoi, lansează în execuție programul `cardmgr`, care se ocupă cu toate introducerile și extragerile de plăci. Acest program `cardmgr` vă permite să „schimbați la cald” plăcile PC Card, astfel încât puteți introduce sau extrage o placă oricând doriți.

■ `/etc/sysconfig/pcmcia` conține un număr de variabile folosite de script-ul `/etc/rc.d/init.d/pcmcia`. În particular, programul Card Services este activat numai dacă linia PCMCIA = yes apare în fișierul `/etc/sysconfig/pcmcia`. Dacă sistemul dumneavoastră nu are nici un dispozitiv PCMCIA, veți găsi linia PCMCIA = no în acel fișier.

Puteți activa programul Card Services la inițializarea sistemului, asigurându-vă că linia PCMCIA = yes apare în fișierul `/etc/sysconfig/pcmcia`. Bineînțeles, acest lucru este de dorit numai dacă calculatorul dumneavoastră are un soclu PCMCIA (fapt adevărat pentru majoritatea laptop-urilor).

Folosirea programului cardctl

Dacă programul PCMCIA rulează, puteți folosi programul `/sbin/cardctl` pentru a monitoriza și controla un soclu PCMCIA. Pentru a vedea starea unui soclu PCMCIA, tastați

```
/sbin/cardctl status
```

Mesajul emis trebuie să prezinte indicatorii de stare ai soclului curent.

Programul `cardctl` are multe argumente. Cu unele din ele puteți suspenda și reactiva o placă și puteți vedea parametrii de configurare, cum ar fi de exemplu întreruperile și regiștrii de configurare. Puteți învăța mai multe despre `cardctl` din manualul online; tastați `man cardctl` pentru a vizualiza pagina din manual.

Folosirea plăcilor PC Card cu suport Linux

În directorul de documentație al programului PCMCIA Card Services, veți găsi un fișier denumit `SUPPORTED.CARDS` (puteți intra în acel director tastând `cd /usr/doc/pcmcia*`). Acest fișier prezintă toate plăcile PCMCIA cunoscute ca fiind compatibile cu cel puțin un sistem.

Lista plăcilor cu suport Linux este prea numeroasă pentru a o include aici. Categoriile de plăci suportate de Card Services sunt următoarele:

- **Plăci Ethernet:** Pot fi utilizate aproape o sută de modele diferite de plăci Ethernet, inclusiv modelele populare 3Com 3c589, 3c589B, 3c589C, 3c589D, Megahertz XJ10BT, CC10BT și Xircom CreditCard CE2.
- **Plăci Fast Ethernet (10/100BASE-T):** Pot fi utilizate plăcile Linksys EtherFast 10/100 și Xircom CreditCard CE3 Fast Ethernet.
- **Plăci Token Ring:** Pot fi utilizate plăcile IBM Token Ring Adapter și 3Com 3c689 TokenLink III.
- **Plăci pentru rețele fără cablu:** Pot fi utilizate AT&T GIS / NCR WaveLAN versiunea 2.0, DEC RoamAbout/DS și Xircom CreditCard Netwave.
- **Plăci modem și seriale:** toate plăcile modem și port serial ar trebui să fie suportate. Singurele excepții sunt modemurile precum Compaq 192, New Media WinSurfer și cele care necesită drivere speciale Windows.
- **Plăci de memorie:** Toate plăcile de memorie SRAM ar trebui să fie compatibile Linux.
- **Plăci adaptor SCSI:** Pot fi utilizate mai mult de 30 de modele diferite de plăci SCSI, inclusiv modele precum Adaptec APA-1460, APA-1460A, APA-1450A Slim SCSI, IBM SCSI, Iomega ZIP Card, NEC PC-9801N-J03R și Toshiba NWB0107ABK, SCSC200B.
- **Plăci pentru unități de disc ATA/IDE:** Pot fi utilizate toate PC Cards ATA/IDE pentru unități de disc.
- **Plăci adaptor ATA/IDE CD-ROM:** Pot fi utilizate câteva adaptoare CD-ROM ATA/IDE, inclusiv Argosy CD-ROM EIDE, Caravelle CD-36N, Creative Technology CD-ROM, Digital Mobile Media CD-ROM, câteva modele EXP și câteva modele I/O-DATA.
- **Plăci multifuncționale:** Pot fi utilizate câteva plăci multifuncționale, inclusiv 3Com 3c562, 3c562B/C/D, 3c563B/C/D, IBM Home și Away Card, Linksys LANmodem 28.8, 33.6, Megahertz EM1144, EM3288, EM3336, Motorola Mariner și Marquis, Ositech Jack of Diamonds și modelele Xircom CreditCard CEM28, CEM33, CEM56.
- **Alte plăci:** Placa Trimble Mobile GPS poate fi utilizată prin intermediul driverului serial/modem.

Pentru lista actualizată de plăci ce pot fi utilizate ar trebui să consultați fișierul `SUPPORTED.CARDS` din versiunea programului PCMCIA Card Services instalată pe sistemul dumneavoastră.

Rețineți că plăcile combo Xircom CE-10BT Ethernet și CE II Ethernet/Modem nu pot fi utilizate.

Informații suplimentare

Pentru a învăța mai multe despre pachetul de software pentru PCMCIA Card Services ar trebui să consultați documentul PCMCIA-HOWTO. Puteți citi acest document tastând `cd /usr/doc/HOWTO` pentru a intra în directorul în care este localizat. Apoi tastați `zmore PCMCIA-HOWTO.gz` pentru a vedea conținutul fișierului PCMCIA-HOWTO. Acest fișier conține cele mai recente informații despre programul Card Services, incluzând probleme frecvent întâlnite și sugestii pentru remediarea acestora.

În particular, ar trebui să parcurgeți fișierul PCMCIA-HOWTO pentru a găsi informații referitoare la placa dumneavoastră PC Card.

Rezumat

Dezvoltată ca o modalitate de atașare a plăcilor de memorie la laptop-uri, plăcile PC Card au devenit o metodă frecvent folosită de a adăuga noi facilități acestor sisteme. Dacă folosiți Linux pe un laptop, aveți nevoie de o modalitate de a accesa plăcile PC. Programul PCMCIA Card Services creat de Dave Hinds furnizează o modalitate de a folosi plăcile PC Card sub Linux. Acest capitol descrie modalitatea de activare a acestui program.

Din lectura acestui capitol, ați învățat următoarele:

- ▶ Plăcile PC Card au fost la început plăci de memorie de mărimea unei cărți de credit care se introduceau într-un slot situat pe una din laturile unui laptop și furnizau un mediu de stocare a datelor.
- ▶ Asociația PCMCIA a standardizat diferite caracteristici ale plăcilor PC, inclusiv interfața electrică, dimensiunile plăcilor și mărimea sloturilor.
- ▶ În prezent, PC Cards sunt frecvent folosite în laptop-uri ca slot de extensie pentru multe dispozitive, ca de exemplu adaptoare de rețea, fax/modemuri, controllere SCSI, plăci de sunet și chiar hard-discuri.
- ▶ Există trei tipuri de plăcile PC Card: tipul I, II și III. Toate plăcile au aceeași lungime și lățime (54 mm x 85,6 mm) – mărimea unei cărți de credit standard. Cele trei tipuri diferă în grosime. Tipul I este cel mai subțire (3,3 mm), iar tipul III este cel mai gros (10,5 mm).
- ▶ Linux Red Hat include programul lui David Hinds PCMCIA Card Services. Cea mai recentă versiune a acestui software este disponibilă via FTP, în directorul `/pub/Linux/kernel/pcmcia` la adresa `sunsite.unc.edu`.
- ▶ Veți găsi o listă cu plăci PC Cards cu suport Linux în fișierul `SUPPORTED.CARDS`.
- ▶ Pentru cele mai recente informații despre un anumit dispozitiv PCMCIA, consultați fișierul `PCMCIA-HOWTO` din directorul `/usr/doc/HOWTO`.

- ▶ Capitolul 18: Rețele dial-up în Linux
- ▶ Capitolul 19: Configurarea unui calculator gazdă Linux pentru Internet
- ▶ Capitolul 20: Rularea unui server World Wide Web în Linux
- ▶ Capitolul 21: Administrarea unei afaceri cu Linux
- ▶ Capitolul 22: Programarea în Linux
- ▶ Capitolul 23: Programarea X în Linux
- ▶ Capitolul 24: Procesarea textului în Linux

Capitolul 18 Rețele dial-up în Linux

În acest capitol

- ▶ Veți înțelege lucrul în rețea
- ▶ Veți înțelege protocoalele SLIP și PPP
- ▶ Veți stabili o conexiune SLIP
- ▶ Veți configura PPP pe calculatorul dumneavoastră Linux
- ▶ Veți folosi PPP pentru transportul de date în rețelele TCP/IP
- ▶ Veți vedea cum se face rutarea TCP/IP într-o conexiune PPP
- ▶ Vă veți transforma sistemul Linux într-un server PPP

Secțiunile anterioare ale acestei cărți sunt dedicate instalării și setării sistemului Linux pe PC-ul dumneavoastră. Ați învățat cum să instalați Linux, să configurați X Window System XFree86 și cum să setați Linux pentru a putea folosi diferite componente hardware. Această parte a cărții se concentrează asupra aspectelor amuzante din Linux (și, de asemenea, asupra unor chestiuni mai serioase). Cele șapte capitole din această parte vă arată cum să conectați PC-ul dumneavoastră Linux la Internet, cum să setați diferite servicii Internet, inclusiv un server World Wide Web, cum să dezvoltați aplicații software și chiar să conduceți o mică afacere cu Linux.

Acest capitol se ocupă de rețelele dial-up – unul dintre primii pași pe care mulți utilizatori trebuie să-i execute pentru a-și conecta calculatorul Linux (și poate propria rețea locală) la Internet.

Dacă aveți un sistem Linux acasă (sau într-un mic birou), veți folosi probabil un modem pentru a vă conecta la Internet. La celălalt capăt al modemului aveți nevoie de un sistem care este deja conectat la Internet. Acest sistem se poate găsi la biroul dumneavoastră, la universitate sau

poate aparține unui furnizor de servicii Internet (Internet Service Provider-ISP). Aceasta este o *rețea dial-up* –, în speță stabilirea unei legături de rețea între calculatorul dumneavoastră Linux și un calculator de la distanță printr-un modem destinat stabilirii legăturilor telefonice.

Acest capitol descrie facilitățile rețelelor dial-up în Linux, cu accent pe protocolul Point-to-Point (PPP) ca metodă de transfer de pachete în rețea prin intermediul unei conexiuni realizate prin apel telefonic.



Referință
incrucișată

În paralel cu lectura acestui capitol, ar trebui să consultați capitolul 16 pentru a vă reaminti noțiuni precum lucrul în rețea, TCP/IP, Internet și Request for Comments (RFC).

Noțiuni elementare privind rețelele dial-up

Rețelele dial-up se referă la conectarea unui PC la o rețea aflată la distanță printr-un modem dial-up. Există o diferență semnificativă între rețelele dial-up și comunicarea serială. Ambele metode folosesc un modem pentru apela un alt calculator prin intermediul liniilor telefonice și a stabili o cale de comunicație, dar programul software pentru comunicație serială (ca de exemplu *minicom*, descris în Capitolul 15) determină calculatorul dumneavoastră să se comporte exact ca un terminal conectat la un calculator de la distanță. Conexiunea dial-up este folosită exclusiv de către programul de software de comunicație serială. Nu puteți să rulați încă o copie a programului de comunicație și să folosiți aceeași conexiune prin modem, de exemplu.

În rețelele dial-up, rulați TCP/IP sau alt protocol de rețea pe calculatorul dumneavoastră precum și pe sistemul de la distanță cu care PC-ul dumneavoastră are o cale de comunicație dial-up. Această cale de comunicație pur și simplu formează unul din nivelurile modelului de rețea pe 7 niveluri OSI. Protocoalele de rețea schimbă pachete de date printr-o conexiune dial-up. Puteți folosi simultan un număr nelimitat de aplicații de rețea pentru a comunica prin singura conexiune dial-up. Folosind rețelele dial-up, calculatorul dumneavoastră devine parte a rețelei din care provine calculatorul de la distanță. (Dacă calculatorul de la distanță nu face parte dintr-o rețea, atunci conexiunea dial-up creează o rețea compusă din calculatorul dumneavoastră și cel de la distanță). Astfel, puteți rula simultan un număr nelimitat de aplicații de rețea, de la un browser de Web până la o sesiune *telnet*, iar toate aplicațiile partajează facilitățile de transport fizic de date ale conexiunii dial-up.

Acest capitol descrie TCP/IP într-o conexiune dial-up, deoarece TCP/IP este protocolul dominant pentru Internet, iar Linux are suport înglobat pentru TCP/IP. Mai exact, acest capitol se referă la TCP/IP într-o legătură cu comunicație serială punct la punct. „Dial-up” înseamnă pur și simplu că majoritatea utilizatorilor folosesc un modem pentru a stabili o legătură de comunicație punct la punct cu un calculator de la distanță.

Ca și TCP/IP în Ethernet, TCP/IP într-o legătură dial-up este o chestiune de specificare a *protocolului* (adică a convenției) de „împachetare” a unui pachet de rețea printr-o legătură de comunicație. Există două protocoale populare pentru TCP/IP într-o legătură de comunicație serială punct la punct:

- Serial Line Internet Protocol (SLIP) este un protocol simplu ce specifică modul de trimitere sub formă de cadre a unui pachet IP pe o linie serială. SLIP este descris în RFC 1055.
- Point-to-Point Protocol (PPP) este un protocol mai avansat pentru stabilirea unei conexiuni TCP/IP în orice legătură punct la punct, inclusiv legături seriale dial-up. PPP este descris în RFC 1661.

Mai întâi vom prezenta o imagine de ansamblu despre SLIP și PPP; apoi vom descrie modul de utilizare a protocoalelor SLIP și PPP pentru a stabili o conexiune de rețea cu un calculator de la distanță.

Protocolul SLIP (Serial Line Internet Protocol)

SLIP a fost la origine un protocol simplu pentru trimiterea sub formă de cadre a unui *pachet IP*, care constă dintr-un antet (header) IP (care include adresele IP sursă și destinație) urmat de date (datele trimise de la sursă la destinație). Documentul RFC 1055, „A Nonstandard for Transmissions of IP Datagrams over Serial Lines: SLIP,” Ronkey, 1988, descrie protocolul SLIP. Așa cum sugerează și titlul documentului RFC 1055 (Un nonstandard pentru transmiterea datagramelor IP prin linii seriale: SLIP), SLIP nu este un standard oficial Internet; este un standard *de facto*.

SLIP definește două caractere speciale pentru trimiterea sub formă de cadre a pachetelor IP:

- SLIP-END este 300 în baza 8 (192 în baza 10) și marchează sfârșitul unui pachet IP.
- SLIP-ESC este 333 în baza 8 (219 în baza 10) și este folosit pentru a „scăpa” (escape) orice caractere SLIP-END sau SLIP-ESC înglobate în pachet (pentru a garanta, de exemplu, că un pachet nu se termină prematur deoarece pachetul IP conține întâmplător un bait cu numărul zecimal 192).

Protocolul presupune trimiterea băiților pachetului IP unul câte unul și marcarea sfârșitului pachetului cu un caracter SLIP-END. Următoarea convenție este folosită pentru a manevra caracterele SLIP-END și SLIP-ESC care se găsesc într-un pachet IP:

- Înlocuiți un caracter SLIP-END cu SLIP-ESC, urmat de 334 în baza 8 (220 în baza 10).
- Înlocuiți un caracter SLIP-ESC cu SLIP-ESC, urmat de 335 în baza 8 (221 în baza 10).

Și atât! Având la bază cele mai populare implementări ale SLIP de la Berkeley UNIX, SLIP mai folosește câteva convenții:

- Pachetele încep și se termină cu caracterul SLIP-END, pentru a asigura că fiecare pachet IP începe într-un mod nou.
- Mărirea totală a pachetului IP (inclusiv antetul IP și datele, dar fără caracterele SLIP pentru trimiterea pachetelor sub formă de cadre) este de 1006 băiți.

Simplitatea protocolului SLIP a dus la popularitatea sa (cu toate că PPP este mai folosit în prezent). SLIP are, totuși, câteva neajunsuri:

- Ambele capete ale conexiunii SLIP trebuie să-și cunoască propria adresă IP, precum și pe cea a celuilalt capăt. Cu toate că au fost concepute câteva scheme care să permită atribuirea dinamică a adreselor IP, protocolul nu are resurse pentru negocierea adresei.
- Ambele capete ale conexiunii SLIP trebuie să folosească aceeași mărime de pachet, pentru că protocolul nu permite celor două capete să negocieze mărirea pachetului.
- SLIP nu are facilități pentru comprimarea datelor. (După cum veți învăța mai târziu în această secțiune, SLIP comprimat (Compressed SLIP) sau CSLIP, introduce comprimarea în SLIP).
- Nu există nici o modalitate de a identifica tipul pachetului în SLIP. În consecință, SLIP poate opera doar cu un singur protocol: acela cu ambele capete cablate. Un mecanism de transport precum SLIP ar trebui să fie capabil de a transporta pachete caracteristice oricărui tip de protocol.



Observație

Lipsa comprimării datelor în SLIP a fost rezolvată de Compressed SLIP (CSLIP), care este descris în RFC 1144, „Comprimarea anteturilor TCP/IP pentru legături seriale de viteză mică” (Compressing TCP/IP Headers for Low-Speed Serial Links), Jacobson, 1990. CSLIP comprimă informația din antetul TCP/IP, care tinde să se

repete în pachetele schimbate între cele două extremități ale conexiunii SLIP. CSLIP nu comprimă datele din pachet.

Deseori CSLIP este denumită *comprimarea Van Jacobson*, după numele autorului său. În PPP se poate lucra cu comprimarea Van Jacobson a anteturilor TCP/IP.

Protocolul PPP (Point-to-Point Protocol)

Protocolul PPP remediază neajunsurile SLIP și definește un protocol mai complex. Spre deosebire de SLIP, PPP este un standard oficial Internet, fiind documentat în RFC 1661, „The Point-to-Point Protocol,” Simpson, 1994 (actualizat în RFC 2153, „PPP Vendor Extensions,” Simpson, Mai 1997).

Protocolul PPP conține următoarele componente principale:

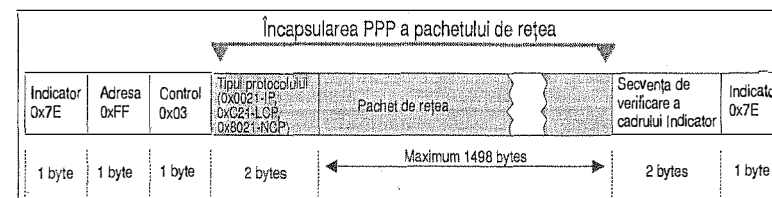
- Un mecanism de grupare în cadre a pachetelor, care folosește o versiune modificată a bine cunoscutului protocol de control la nivel înalt al legăturii de date (High-Level Data Link Control – HDLC).
- Un protocol de control al legăturii (Link Control Protocol) pentru a stabili, configura și testa legătura de date.
- Un protocol de control al rețelei (Network Control Protocol – NCP) care permite PPP să transporte mai multe tipuri de pachete de rețea – ca de exemplu IP, IPX și NetBEUI (Network BIOS Extended User Interface) – în aceeași conexiune.

PPP înlocuiește treptat SLIP ca protocol de transport al pachetelor prin legăturile punct cu punct. Pe lângă legăturile seriale omniprezente, câteva versiuni ale PPP lucrează cu alte câteva tipuri de legături punct cu punct. Câteva dintre legăturile point-to-point cu care lucrează PPP includ Frame Relay, SONET/SDH (Synchronous Optical Network – Rețea optica sincrona/Synchronous Digital Hierarchy – Ierarhie digitală sincronă), X.25 și ISDN (Integrated Services Digital Network – Rețea cu servicii digitale integrate).

Un cadru PPP are o structură mai complexă decât SLIP. Structura cadrului PPP este bazată pe standardul ISO 3309, „Data Communications – High-Level Data Link Control Procedures – Frame Structure”, 1979. Protocolul HDLC (High-Level Data Link Control) folosește un caracter indicator special pentru a marca începutul și sfârșitul unui cadru. Figura 18-1 prezintă structura unui cadru complet PPP.

Figura 18-1

Formatul unui cadru PPP



Așa cum arată figura 18-1, cadrele PPP încep și se termină cu un caracter indicator a cărui valoare este întotdeauna 0x7E (adică 7E în notație hexazecimală). Câmpurile Adresă și Control provin de la HDLC; ele au valorile fixe 0xFF și respectiv 0x03. Datele PPP sunt formate dintr-un câmp ce conține un protocol pe doi băiți. (De fapt, acest câmp poate avea doar un singur bait; lungimea câmpului protocolului este negociată cu protocolul de control al legăturii – Link Control Protocol).

În interiorul pachetului de rețea încapsulat, PPP folosește 0x7D drept caracter Escape. Pentru a trimite un bait ce are o semnificație specială (ca de exemplu 0x7E, care marchează începutul și sfârșitul unui cadru), PPP folosește următorii pași:

Înglobează 0x7D în date

Plasează baitul de date „scăpat” (escaped)

Comută pe valoarea complementară al șaselea bit al celui bait de date

Astfel, dacă datele dintr-un pachet PPP includ 0x7E, acel bait este înlocuit de secvența de doi baiți 0x7D, urmată de 0x5E. (În cazul în care comutați pe valoarea complementară al șaselea bit al baitului 0x7E sau 0111 1110 în baza 2, obțineți 0x5E sau 0101 1110 în baza 2).

Când folosiți protocolul PPP pentru a seta o legătură între PC-ul dumneavoastră Linux și un calculator aflat la distanță, PC-ul dumneavoastră trimite mai întâi pachete LCP pentru a seta legătura datelor. După ce legătura de date fizică este stabilă și parametrii opționali sunt negociați, PC-ul dumneavoastră trimite pachete NCP pentru a selecta unul sau mai multe protocoale de rețea care să fie folosite pentru legătura respectivă. Apoi, oricare dintre aceste protocoale de rețea pot trimite pachete printr-o legătură PPP.



Sugestie

Nu trebuie să știți toate detaliile PPP pentru a-l putea folosi în mod operativ. Secțiuni ulterioare ale acestui capitol descriu cum puteți folosi PPP pentru a stabili o conexiune de rețea TCP/IP cu alt calculator.

Realizarea unei conexiuni SLIP

Pentru a realiza o conexiune SLIP între PC-ul dumneavoastră Linux și un sistem de la distanță, ambele sisteme trebuie să poată lucra cu SLIP. Ambele capete trebuie să ruleze câteva aplicații SLIP în intervalul de funcționare a conexiunii.

Pentru a include SLIP în nucleul dumneavoastră Linux, ar trebui să acceptați răspunsurile prestabilite la următoarele întrebări din timpul configurării nucleului (pentru a realiza acest lucru tastați **make config**):

```
SLIP (serial line) support ( CONFIG_SLIP ) [M/n/y/?] y
CSLIP compressed headers ( CONFIG_SLIP_COMPRESSED ) [y/n/?]
Keepalive and linefill ( CONFIG_SLIP_SMART ) [y/n/?]
Six bit SLIP encapsulation ( CONFIG_SLIP_MODE_SLIP6 ) [N/y/?]
```



Referință încrucișată

Capitolul 2 descrie modul de configurare și regenerare a nucleului. Aveți nevoie de asemenea și de suport TCP/IP în nucleu.

De obicei, primul pas în stabilirea unei conexiuni SLIP este formarea numărului de telefon al sistemului de la distanță, intrarea în sistem (login) și pornirea SLIP de pe acel sistem. Apoi trebuie să porniți SLIP pe calculatorul dumneavoastră. După ce la cele două capete ale conexiunii a fost lansat SLIP, atunci conexiunea SLIP este activă. Pentru a folosi această conexiune, trebuie să configurați de asemenea interfața (cu ajutorul comenzii **ifconfig**) și să setați rutarea (cu comanda **route**) astfel încât pachetele de rețea trimise de la sistemul dumneavoastră să poată ajunge la destinație. Puteți face toate aceste lucruri cu ajutorul unui program numit **dip** (Dial-up IP Protocol Driver), care automatizează procesul de setare a conexiunii SLIP. În următoarele secțiuni veți învăța cum să folosiți comanda **dip**.

Verificarea suportului SLIP

Înainte de a încerca SLIP, ar trebui să verificați dacă nucleul dumneavoastră include suport pentru utilizarea acestuia. Tastați **dmesg | more** pentru a vedea mesajele de inițializare. Dacă nucleul dumneavoastră include suport SLIP, pe ecran vor apărea următoarele mesaje:

```
CSLIP: code copyright 1989 Regents of the University of California
SLIP: version 0.8.4-NET3.019-NEWTTY-MODULAR (dynamic channels, max=256).
SLIP linefill/keepalive option.
```

Primele două linii se referă la suportul CSLIP și respectiv SLIP. Dacă vedeți aceste linii, puteți considera că nucleul dumneavoastră are include suport pentru SLIP.

Dacă suportul SLIP se află sub forma unui modul încărcabil, ar trebui să-l încărcați prin conectarea ca **root** și tastarea comenzii:

```
modprobe slip
```

Tastați **lsmod** pentru a verifica dacă modulul SLIP a fost încărcat.

Obținerea informațiilor despre sistemul de la distanță

Mecanismul exact al stabilirii unei conexiuni SLIP cu un sistem aflat la distanță depinde de setarea celui sistem. Dacă ați obținut un cont SLIP de la un furnizor de servicii Internet (Internet Service Provider – ISP), atunci acesta ar trebui să vă dea informațiile de care aveți nevoie pentru a stabili o conexiune SLIP. Informațiile trebuie să conțină cel puțin următoarele:

- Numărul de telefon pe care trebuie să-l formați pentru a vă conecta la calculatorul de la distanță.
- Numele de utilizator și parola pe care trebuie să le folosiți pentru a vă conecta la sistem.
- Adresa IP pentru ambele capete ale conexiunii SLIP. Mulți furnizori de servicii Internet furnizează o adresă IP fixă (statică), în timp ce alte ISP-uri atribuie adrese IP în mod dinamic. Obțineți, de asemenea, adresele IP ale serverelor de nume (un server de nume convertește numele în adrese IP) ale unui server poștal și al unuia de mesaje. Aceste adrese IP sunt necesare atunci când începeți să folosiți aplicații Internet pe sistemul dumneavoastră; nu aveți nevoie de ele în timpul setării conexiunii SLIP.

De obicei, un furnizor de servicii Internet setează un cont SLIP astfel încât acel program SLIP pornește automat după ce intrați în sistem. Furnizorul de servicii Internet pornește automat SLIP cu specificația ca aplicația SLIP să ruleze în locul unui program shell (ca de exemplu **/bin/bash**) pentru acel nume de cont. Un utilizator obișnuit poate avea o intrare în fișierul **/etc/passwd** care arată ca în exemplul următor:

```
naba:ZbW0xq2xGA03g:501:100:Naba Barkakati:/home/naba:/bin/bash
```

Semnul (**:**) separă câmpurile în această linie din fișierul **/etc/passwd**. Primul câmp este numele utilizatorului; ultimele două câmpuri sunt directorul și respectiv programul shell (programul care rulează după ce utilizatorul intră în sistem). Pentru un cont SLIP, intrarea poate fi următoarea:

```
slipxxx:ZbW0xq2XfL03g:501:101:SLIP user xx:/tmp:/usr/bin/pp1
```

Numele utilizatorului poate fi oarecum ciudat (**slipxxx**), directorul inițial poate fi **/tmp**, iar ultimul câmp conține numele programului SLIP de pe sistemul de la distanță. Pe o stație de lucru HP, de exemplu, **/usr/bin/pp1** este aplicația software de legătură punct cu punct ce stabilește o conexiune SLIP.

Dacă sistemul de la distanță se află în biroul dumneavoastră, puteți avea un login SLIP separat sau puteți intra în sistem ca un utilizator obișnuit și apoi porniți aplicația SLIP. Puteți automatiza această parte a procesului folosind programul **dip** pe PC-ul dumneavoastră Linux.

Folosirea programului dip pentru stabilirea unei conexiuni SLIP

Programul dip (Dial-up IP Protocol Driver) vă permite să automatizați pașii necesari pentru setarea unei conexiuni SLIP cu un calculator de la distanță. Dacă doriți, puteți rula dip în mod interactiv. Totuși, cel mai bun mod de a folosi dip este de a pregăti un *fișier script* (un fișier text conținând comenzi pentru dip) și apoi să rulați dip cu acel fișier script ca parametru de intrare. Modul de configurare a unui script dip va fi prezentat în secțiunea „Setarea unui script dip”. În orice caz, mai întâi încercați să folosiți interactiv câteva comenzi dip.

Rularea interactivă a scriptului dip

Pentru a vă familiariza cu comenzile dip, rulați-le interactiv. Rulați dip cu opțiunea -t, după cum urmează:

```
/usr/sbin/dip -t
DIP: Dialup IP Protocol Driver version 3.3.70-uri (8 Feb 96)
Written by Fred N. van Kempen, MicroWalt Corporation.
```

```
DIP> help
DIP knows about the following commands:
```

beep	bootp	break	chatkey	config
databits	dec	default	dial	echo
flush	get	goto	help	if
inc	init	mode	modem	netmask
onexit	parity	password	proxyarp	print
psend	port	quit	reset	send
shell	skey	sleep	speed	stopbits
term	timeout	wait		

```
DIP> quit
```



Sugestie Când rulați dip interactiv, puteți folosi comanda help pentru a examina lista comenzilor acceptate de acesta. Dacă aveți toate informațiile, puteți să folosiți dip interactiv și să invocați comenzile corespunzătoare pentru a forma numărul de telefon al calculatorului de la distanță, să executați login în sistemul de la distanță și să stabiliți conexiunea SLIP. Tabelul 18-1 conține comenzile dip disponibile, cu sintaxa și semnificația fiecăreia. În mod normal aceste comenzi se folosesc într-un fișier script dip.

Tabelul 18-1 Comenzile dip

Comandă	Semnificație
beep [<i>ori</i>]	Emite un sunet (bip) de un anumit număr de ori.
bootp	Folosește protocolul BOOTP pentru a determina adresa locală IP. (BOOTP sau Bootstrap Protocol, definit în RFC 951, furnizează unei stații de lucru modalitatea de a-și determina propria adresă IP. Protocolul a fost dezvoltat pentru a permite stațiilor de lucru fără hard-disc de a găsi un calculator gazdă din care să preia și să execute un fișier de inițializare.
break	Trimite o pauză (BREAK). (Consultați Capitolul 15 pentru explicații.)
chatkey keyword [<i>value</i>]	Adaugă un cuvânt cheie (și o valoare asociată) la lista răspunsurilor modemului pe care dip le recunoaște. chatkey RING 10, de exemplu, asociază valoarea 10 cu răspunsul RING al modemului.

Comandă	Semnificație
config [interface routing] [pre] up down post] arguments	Folosește interfața specificată și parametrii de configurare din tabela de rutare. În mod prestabilit, această comandă este dezactivată.
databits 7 8	Setează numărul biților de date la 7 sau 8 (trebuie să specificați una din cele două valori). Numărul biților de date se mai numește și <i>lungimea cuvântului</i> în comunicațiile seriale.
dec \$variable [<i>number</i> \$var2]	Decrementează valoarea specificată (pentru a denumi o variabilă, dip folosește un nume care are semnul dolar ca prefix). Puteți introduce un număr sau altă variabilă pentru a indica valoarea ce trebuie diminuată. Dacă nu introduceți un număr sau o a doua variabilă, dip reduce valoarea variabilei cu 1.
default	Setează direcția prestabilită către calculatorul de la distanță cu care dip a stabilit conexiunea. Ruta prestabilită este folosită pentru a trimite pachete către orice rețea a cărei adresă este necunoscută.
dial phonenum [<i>num_secunde</i>]	Formează numărul de telefon specificat și așteaptă numărul de secunde indicat pentru a primi un răspuns. Dacă nu specificați un număr, dip face o pauză de 60 de secunde. După ce invocă comanda de formare a numărului de telefon adresată modemului, dip analizează răspunsul modemului și setează variabila numită \$errlvl conform răspunsului (0=OK; 1=CONNECT).
echo on off	Comanda echo on activează afișarea tuturor răspunsurilor modemului și trimiterea textului către modem. (Această comandă vă poate ajuta să depanați scripturi dip.) Comanda echo off dezactivează această caracteristică și determină dip să lucreze în mod silențios. Deoarece îmi place întotdeauna să văd ce se întâmplă, eu folosesc echo on în scripturile dip.
exit [<i>status</i>]	Iese din scriptul dip cu codul de stare specificat (lăsând conexiunea SLIP curentă intactă și programul dip în funcțiune). În acest fel trebuie să ieșiți dintr-un script dip după ce ați stabilit o conexiune SLIP corectă.
flush	Șterge intrarea ce a fost citită de la modem sau de la terminal.
get \$variable [<i>number</i> ask remote [<i>timeout</i>] \$var2]	Obține valoarea unei variabile specificate. Folosiți cuvântul cheie ask pentru a solicita utilizatorul să introducă o valoare. Folosiți un număr pentru a seta pur și simplu variabila la acea valoare. Folosiți cuvântul cheie remote pentru a citi o valoare din calculatorul de la distanță. Puteți introduce, de asemenea, numele unei a doua variabile ca sursă a valorii.
goto label	Face un salt la eticheta (label) specificată din script. Eticheta marchează o locație în fișierul script. Sintaxa unei etichete folosește textul acesteia urmat de două puncte (:). Veți vedea un exemplu în acest sens în secțiunea „Setarea unui script dip”.
help	Tipărește informații de asistență (folosite interactiv).
if expresie goto eticheta	Testează <i>expresie</i> și face un salt la eticheta dacă expresia este adevărată. Expresia este de forma \$variabila operator constant, în care \$variabila desemnează o variabilă dip iar operator este ==, !=, <, >, <= sau >=. constant este o valoare constantă.
inc \$variabila [<i>number</i> \$var2]	Incrementează valoarea specificată. Puteți indica un număr sau altă variabilă ca valoare de incrementare. Dacă nu indicați un număr sau o a doua variabilă, dip incrementează valoarea variabilei cu 1.

(continuare)

Tabelul 18-1 Continuare

Comandă	Semnificație
init <i>sir_init</i>	Specifică șirul de inițializare ce trebuie trimis la modem înainte ca <i>dip</i> să execute comanda <i>dia1</i> . Valoarea prestabilită pentru <i>sir_init</i> este <i>ATEO QO V1 X1</i> . (Consultați Capitolul 15 pentru o listă de comenzi ale modemurilor Hayes.)
mode SLIP CSLIP PPP TERM	Stabilește protocolul folosit pentru conexiune. Protocolul prestabilit este SLIP. Dacă furnizorul dumneavoastră de servicii Internet poate lucra cu Compressed SLIP, atunci folosiți acest mod.
modem <i>tip_modem</i>	Specifică tipul modemului. Singura valoare acceptabilă este HAYES, pe care <i>dip</i> o consideră prestabilită.
netmask <i>xxx.xxx.xxx.xxx</i>	Specifică masca rețelei ce va fi folosită în configurarea interfeței SLIP. (Consultați Capitolul 16 pentru o discuție asupra terminologiei TCP/IP, între care masca de rețea.)
parity E O N	Stabilește paritatea ce trebuie folosită pentru comunicații seriale. Folosiți <i>E</i> pentru par (even), <i>O</i> pentru impar (odd) și <i>N</i> pentru nici o paritate.
password	Solicită utilizatorului o parolă, pe care apoi o trimite la modem.
proxyarp	Stabilește ARP-ul proxy. (Descris în RFC 1027, ARP-ul proxy permite unui calculator gazdă – ce trebuie să fie și poartă de acces – să reprezinte alte calculatoare gazdă.
print <i>orice_text</i> [<i>\$variabila</i>]	Afișează textul și valorile oricărei variabile.
port <i>nume_dispozitiv</i>	Specifică numele dispozitivului portului serial la care este conectat modemul. Trebuie să indicați numele dispozitivului fără prefixul <i>/dev/</i> . Astfel, folosiți <i>cua0</i> dacă modemul este pe COM1.
quit	Iese din programul <i>dip</i> cu starea nonzero.
reset	Trimite la modem șirul <i>+++</i> , urmat de <i>ATZ</i> . (Cele trei semne plus atrag atenția modemului, iar <i>ATZ</i> resetează modemul.)
securidf <i>fixed_part</i>	Stochează partea fixă a unei parole SecureID.
securid	Solicită utilizatorului partea variabilă a parolei generată de o placă ACE System SecureID.
send text [<i>\$variabila</i>]	Trimite către modem textul și valorile oricărei variabile.
skey [<i>timeout</i> <i>\$variabila</i>]	Determină <i>dip</i> să caute un semnal S/Key de la calculatorul aflat la distanță. (S/Key este un sistem de autentificare dezvoltat de Bellcore.)
sleep <i>num_secunde</i>	Așteaptă numărul de secunde specificat.
speed <i>biti_pe_secunda</i>	Stabilește viteza portului serial. (Indicați un număr în biți pe secundă.)
stopbits 1 2	Stabilește numărul biților de stop pentru comunicația serială.
term	Determină <i>dip</i> să intre în modul emulare de terminal. Acest mod vă permite să interveniți direct prin conexiunea serială. Puteți tasta <i>Ctrl-]</i> pentru a ieși din modul terminal.

Comandă	Semnificație
timeout <i>num_secunde</i>	Setează variabila de pauză (timeout) la un număr specificat de secunde. Dacă în acele secunde nu intervine nici o activitate, <i>dip</i> întrerupe conexiunea și își încheie execuția.
wait text [<i>timeout</i> <i>\$variabila</i>]	Așteaptă să sosească textul specificat. Argumentul opțional indică intervalul de timp de așteptare.

Pe lângă aceste comenzi, *dip* furnizează și câteva variabile interne. Tabelul 18-2 listează variabilele interne ale programului *dip*.

Tabelul 18-2 Variabile interne dip

Variabilă	Descriere
\$errlvl	Codul rezultat al ultimei comenzi executate. Zero indică succes; orice altă valoare indică eroare.
\$local	Numele calculatorului gazdă pentru PC-ul dumneavoastră Linux (sistemul local).
\$locip	Adresa IP a PC-ului dumneavoastră Linux.
\$modem	Numele modemului. (Singura valoare acceptată este HAYES.)
\$mtu	Unitatea maximă de transfer (Maximum Transfer Unit) – numărul maxim de biți pe care-l poate conține un pachet.
\$port	Numele dispozitivului serial (de exemplu <i>cua0</i> pentru COM1) folosit pentru conexiunea SLIP.
\$remote	Numele calculatorului gazdă pentru sistemul de la distanță – celălalt capăt al conexiunii SLIP.
\$rmtip	Adresa IP a capătului de la distanță al conexiunii SLIP.
\$speed	Rata de transfer a datelor între PC-ul dumneavoastră și modem (în biți pe secundă).

Când încercați pentru prima dată o conexiune SLIP, veți dori să rulați *dip* în interactiv și să folosiți comanda *term* pentru a intra în sistem și a stabili legătura. Următorul exemplu prezintă modul de conectare ca *root* pe sistemul meu Linux și de stabilire a unei conexiuni SLIP cu o stație de lucru HP de la distanță. (Datele introduse de autor sunt scrise cu caractere aldine; comentariile sunt scrise cu litere cursive.)

```
Inbsoft: /home/naba/slip# dip -t
DIP: Dialup IP Protocol Driver version 3.3.7o-uri (8 feb 96)
Written by Fred N. van Kempen, Microwalt Corporation.
```

```
DIP> get $locip 140.90.23.195
DIP> get $rmtip 140.90.23.194
DIP> port cua 0
DIP> speed 38400
DIP> term
[ Entering TERMINAL mode. Use CTRL-] to get back ]
```

```
NO CARRIER
ATZ
OK
ATDT5551212 (folositi numarul de telefon corect)
CONNECT 28800/ARQ/V34/LAPM/V42BIS
```

```
GenericSysName [Release] (see/etc/issue)
dialin login: naba (introduceti numele dumneavoastra pentru conectare)
```

Password: (introduceți parola dumneavoastră)

(mesajele irelevante au fost sterse)

You have mail.

TERM = (hp) (apasati Enter)

/users/naba 21 % **ppl**

ppl: starting for naba at Mon Apr 11 20:45:55 1998

/dev/ttyd01 38400 Linet=140.90.23.194 Rinet= 140.90.23.195

Mask=255.255.255.0 SLIP

initialization complete, running protocol

(Apasati **Ctrl-J**)

[Back to LOCAL mode.]

DIP> **default**

DIP> **mode SLIP**



Sugestie

După ultima comandă SLIP, dip își termină execuția, iar conexiunea este stabilită.

Din acest moment pot folosi aplicațiile de rețea, ca de exemplu telnet și ftp, pentru a accesa sistemul de la distanță. Când nu mai am nevoie de conexiunea SLIP, intru în sistem ca root din nou și tasez **dip -k cua0** pentru a întrerupe conexiunea.

Observați că, atunci când lucrați în modul dip interactiv, trebuie să comutați pe modul terminal (cu ajutorul comenzii **term**) pentru a trimite comenzi modemului. Trebuie să fiți în modul terminal când trimiteți comenzile de formare a numărului de telefon către modem și pentru a intra în sistemul de la distanță atunci când vedeți promptul de intrare în sistem (**login**).

După ce introduceți toate comenzile necesare sistemului de la distanță ar trebui să reveniți în modul de comandă dip tastând **Ctrl-J**. Apoi puteți încheia configurarea SLIP introducând comanda **default**, urmată de comanda **mode SLIP**. Comanda **default** determină dip să seteze o direcție prestabilită folosind conexiunea SLIP (care înseamnă că orice pachet de rețea cu adresă necunoscută se îndreaptă către conexiunea SLIP). Comanda **mode SLIP** determină dip să comunice cu capătul aflat la distanță folosind protocolul SLIP.



Sugestie

Dacă furnizorul dumneavoastră de servicii Internet poate lucra cu Compressed SLIP, atunci folosiți comanda **mode CSLIP** în loc de **mode SLIP**.

Setarea unui script dip

Așa cum se vede în secțiunea precedentă, puteți stabili o conexiune SLIP rulând dip în mod interactiv cu opțiunea **-t**. Această soluție nu este totuși convenabilă dacă doriți să stabiliți în mod sistematic o conexiune SLIP. Dacă furnizorul dumneavoastră de servicii Internet vă oferă adrese fixe pentru ambele capete ale conexiunii SLIP, puteți crea un fișier script care să stabilească automat conexiunea. Fișierul script conține comenzi pentru programul dip; trebuie să scrieți fișierul script conform sintaxei cerute de programul dip.



Sugestie

Dacă fișierul dumneavoastră script dip folosit ca parametru de intrare pentru programul dip se numește **connect.dip** (de obicei, extensia dip folosită pentru fișiere script destinate programului dip), atunci puteți tasta dip **connect**. Această comandă determină programul dip să citească și să execute comenzile din fișierul **connect.dip**.

Fișierul script însuși este în primul rând o succesiune de comenzi pentru dip, asemănătoare comenzilor pe care le folosiți când rulați dip cu opțiunea **-t**. Singura diferență majoră este că

trebuie să folosiți comanda **send** pentru a trimite text către modem și comanda **wait** pentru a căuta un anumit text ce se întoarce de la sistemul de la distanță. În continuare este prezentat un fișier script dip pe care autorul îl folosește pentru a se conecta la o stație de lucru HP (Trebuie să introduc manual comanda **ppl** pentru a stabili o legătură point-to-point folosind protocolul SLIP):

```
# Fisier: connect.dip
```

```
#
```

```
# Stabilește o conexiune SLIP dial-up.
```

```
#
```

```
# Naba Barkakati, 4/11/98
```

```
main:
```

```
# Vizualizam toate mesajele emise pentru ca depanarea sa decurga usor
    echo on
```

```
# Setarea adresei mele IP
```

```
    get $locip 140.90.23.195
```

```
# Setarea adresei IP a calculatorului aflat la distanta
```

```
# Observatie: Aceste valori sunt pentru exemplificare -ar trebui sa folositi
```

```
# adresele IP corecte
```

```
    get $rmtip 140.90.23.194
```

```
# Seteaza masca rețelei la s10 to. 255.255.2550
```

```
    netmask 255.255.2550
```

```
# Selecteaza portul serial si viteza (modemul meu este pe COM1 care
```

```
# este /dev/cua0 - selectati-l pe al dumneavoastră)
```

```
    port cua0
```

```
    speed 38400
```

```
# Trimite secventa de initializare la modem
```

```
    send ATZ\r
```

```
    wait OK 2
```

```
    send ATE1M1V1X4L3S0=0\&c1Q0DT5551212\r
```

```
    wait CONNECT 75
```

```
# Verifica „errlvl“
```

```
    if $errlvl != 0 goto error
```

```
# Suntem conectati. Log in.
```

```
login:
```

```
    wait ogin: 30
```

```
if $errlvl != 0 goto no_login_prompt
```

```
# Trimite numele dumneavoastra de utilizator (folositi identificatorul
dumneavoastra pe linia urmatoare)
```

```
    send your-username\r
```

```
    wait word: 10
```

```
# Trimite parola (folositi parola dumneavoastra pe linia urmatoare)
```

```
    send your-password-here
```

```
# Urmatoarea secventa depinde de sistemul dumneavoastra
```

```
# In acest caz, eu raspund la solicitarea sistemului de a introduce numele
terminalului
```

```
# terminal name
```

```
    wait TERM 10
```

```
    send \r
```

```
    sleep 5
```

```

send \r
send \r
wait some-text-in-the-prompt 30

# Porneste SLIP la capatul aflat la distanta
print Sending command to start SLIP...
send ppl\n
wait some-text-indicating-SLIP-is-running 30

default
mode SLIP
# Tipareste un mesaj si isi termina executia
print Connected... $locip -> $rmtip
goto exit

no_login_prompt:
print No login prompt...
goto error

error:
print CONNECT FAILED
quit 1

exit:

```

Nu uitați că trebuie să înlocuiți articolele scrise cu litere cursive cu textul și numerele corespunzătoare situației dumneavoastră. În particular, dacă ați obținut un cont SLIP de la un ISP (furnizor de servicii Internet), trebuie să-i cereți acestuia adresele IP și numărul de telefon. De asemenea, pașii exacți pentru conectarea la un sistem aflat la distanță variază de la un sistem la altul.

Verificarea conexiunii SLIP

După ce conexiunea SLIP este stabilită, puteți folosi comanda `ifconfig` pentru a vedea dacă dispozitivul SLIP `s10` este configurat (`dip` face acest lucru pentru dumneavoastră).

Iată mesajul tipic emis la invocarea comenzii `ifconfig` după ce conexiunea SLIP a fost stabilită și este activă:

```

ifconfig
lo Link encap:Local Loopback
inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING MTU:2000 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:25 errors:0 dropped:0 overruns:0

s10 Link encap:Serial Line IP
inet addr:140.90.23.195 P-t-P:140.90.23.194 Mask:255.255.255.0
UP POINTOPOINT RUNNING MTU:296 Metric:1
RX packets:4 errors:0 dropped:0 overruns:0
TX packets:4 errors:0 dropped:0 overruns:0

eth0 Link encap:10Mbps Ethernet HWaddr 00:20:AF:E8:A2:25
inet addr:206.197.168.200 Bcast:206.197.168.255
Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
RX packets:2087 errors:0 dropped:0 overruns:0
TX packets:543 errors:0 dropped:0 overruns:0
Interrupt:10 Base address:0x210

```



Secret

Suplimentar, dacă folosiți comanda `default`, `dip` adaugă o direcție prestabilită la conexiunea SLIP. Pentru a verifica acest lucru, tastați comanda `/sbin/route` și observați dacă pachetele pentru adresa IP 140.90.23.194 (care, în acest caz, reprezintă capătul de la distanță al conexiunii SLIP) sunt trimise dispozitivului `s10` – dispozitivul conexiunii SLIP. Pachetele pentru destinația `default` (orice adresă IP diferită de cele listate în tabelul de direcționare) ar trebui trimise, de asemenea, adresei IP 140.90.23.194 – capătul aflat la distanță al conexiunii SLIP. Vă sunt necesare aceste intrări în tabela de rutare pentru ca sistemul dumneavoastră Linux să poată comunica cu sistemul de la distanță.

Încheierea unei conexiuni SLIP

Pentru a încheia o conexiune SLIP, trebuie să utilizați comanda `dip` cu opțiunea `-k`, urmată de dispozitivul serial folosit pentru conexiunea SLIP. Intrați în sistem ca `root` și tastați următoarea comandă în directorul `/dev/cua0` pentru a încheia o conexiune SLIP:

```

dip -k cua0
DIP: Dialup IP Protocol Driver version 3.3.70-uri (8 Feb 96)
Written by Fred N. van Kempen, MicroWalt Corporation.

```

DIP: process 951 killed

Conectarea la o rețea de la distanță ca un client PPP

PPP este un protocol mai complex și mai versatil decât SLIP, însă mecanismul conectării la o rețea de la distanță este foarte asemănător cu cel folosit de SLIP. Și în cazul PPP aveți adrese IP pentru cele două capete ale conexiunii PPP, un număr de telefon ce trebuie format și un nume de utilizator cu care să vă conectați la sistemul de la distanță.



Observație

La fel cum programul `dip` vă ajută să automatizați setarea SLIP, alte două programe – `pppd` și `chat` – vă ajută să configurați o conexiune PPP. Înainte de orice însă, trebuie să vă asigurați că nucleul dumneavoastră Linux conține dispozitivele de rețea PPP.

Verificarea suportului PPP

La fel ca la SLIP, trebuie să aveți nucleul configurat cu suport PPP. Atunci când regenerați nucleul Linux folosind comanda `make config` (consultați capitolul 2 pentru detalii), ar trebui să răspundeți afirmativ la următoarea întrebare despre facilitățile PPP:

```

PPP (point-to-point) support (CONFIG_PPP) [M/n/y/?] y

```

Puteți folosi comanda `cat /proc/net/dev` pentru a verifica dacă dispozitivele PPP sunt disponibile pe sistemul dumneavoastră. De obicei, dispozitivul `ppp0` trebuie să apară între datele de ieșire.

Mai mult, când porniți sistemul, mesajele de inițializare arată dacă dispozitivele PPP au fost configurate. Utilizați comanda `dmesg | more` pentru a vedea mesajele de boot, care ar trebui să fie similară cu următoarele rânduri:

```

IP Protocols: IGMP, ICMP, UDP, TCP
PPP: version 2.2.0 (dynamic channel allocation)
TCP compression code copyright 1989 Regents of the University of California
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.

```



Referință încrucișată

Dacă nu observați nici o dovadă a existenței suportului PPP în nucleu, trebuie să regenerați nucleul folosind etapele descrise în capitolul 2. Apoi trebuie să vă asigurați că suportul PPP este activ atunci când executați comanda `make config`.

Adunarea informațiilor pentru o conexiune PPP

Mulți furnizori de servicii Internet oferă acces PPP la Internet printr-unul sau mai multe sisteme administrate de aceștia. Dacă contractați un astfel de serviciu, furnizorul trebuie să vă ofere informațiile de care aveți nevoie pentru a realiza o conexiune PPP cu un calculator de la distanță. Informațiile minimale trebuie să cuprindă următoarele:

- Adresa IP pentru capătul client al conexiunii. (Această adresă IP va fi asociată cu interfața PPP a PC-ului dumneavoastră – portul serial.)
- Adresa IP pentru interfața PPP a furnizorului. (Această adresă poate apărea ca adresa porții de acces, deoarece această interfață este poarta de acces a sistemului dumneavoastră către Internet.)
- Numărul de telefon ce trebuie format pentru a realiza conexiunea cu calculatorul de la distanță.
- Numele utilizatorului și parola cu care vă conectați la calculatorul de la distanță.

Cei mai mulți furnizori de servicii Internet vă pun la dispoziție adresele IP pentru un server de nume, un server pentru poștă și un server de mesaje. Aceste adrese, nu sunt importante în mecanismul de setare a unei conexiuni PPP.

Utilizarea programelor chat și pppd pentru realizarea conexiunii PPP

Pentru a seta o conexiune de rețea PPP între două sisteme, este necesar ca la ambele capete să ruleze aplicații software PPP. De obicei, furnizorul vă oferă un cont care este setat astfel încât aplicația software PPP se lansează automat la intrarea în sistem. În acest caz, tot ce vă mai rămâne de făcut este să lansați programul PPP pe sistemul dumneavoastră după ce executați `login` la sistemul de la distanță.



Observație

Iată două etape de bază în setarea unei conexiuni PPP:

1. Formați numărul de telefon și conectați-vă la sistemul de la distanță. (Voi presupune că, atunci când executați `login`, calculatorul de la distanță lansează automat programul său PPP.)
2. Lansați softul PPP pe sistemul dumneavoastră Linux.

La fel cum programul `dip` poate lucra cu o conexiune SLIP, programul `pppd` se ocupă de comunicarea prin PPP pe o linie dial-up. Prin urmare, după ce executați `login` și lansați programul PPP pe calculatorul de la distanță, trebuie să rulați `pppd` pe sistemul dumneavoastră Linux.



Secret



CD

Numele programului `pppd` vine de la Point-to-Point Protocol Daemon. (În UNIX, termenul *daemon* se referă la un program ce rulează în fundal și execută un proces util.) Programul `pppd` furnizează o opțiune prin care puteți invoca alt program ce stabilește comunicația serială și finalizează deschiderea unei sesiuni de lucru la distanță.

În Linux se folosește de obicei `pppd` cu programul `chat`. Dacă ați instalat Linux de pe CD-ROM-ul anexat acestei cărți, veți găsi ambele programe, `pppd` și `chat`, în directorul `/usr/sbin` din sistemul dumneavoastră Linux.

Utilizarea programului chat

Trebuie să înțelegeți cum funcționează `chat` pentru a-l putea folosi automat la formarea numărului de telefon și conectarea la calculatorul de la distanță. Programul `chat` are rolul de a procesa un fișier script ce folosește un model de text *expect-send* (așteptare-trimitere). Programul caută șirul *expect*, iar când îl primește, `chat` trimite șirul *send*.

Să presupunem că dorim ca `chat` să caute șirul *login*: (care reprezintă solicitarea sistemului UNIX de a introduce numele unui utilizator) și apoi să trimită numele de utilizator al utilizatorului (*naba*). Apoi dorim ca `chat` să aștepte promptul *Password*: și să trimită parola autorului. Pentru aceasta aș folosi următoarea pereche *expect-send*:

```
ogin: naba word: parola-mea
```

Ați observat probabil că am omis litera *l* de la *login* și câteva caractere din promptul *password*; am evitat să fiu prea exact, pentru ca `chat` să reușească chiar dacă textul primit este puțin diferit. În concluzie, primele caractere ale promptului *login* sau ale șirului *Password*: pot fi omise.

Pauze (timeouts)

Când așteaptă un anumit șir de text, `chat` folosește o facilitare de pauză. Dacă șirul nu sosește în perioada de pauză, `chat` se deplasează la următorul șir. Valoarea prestabilită a pauzei este de 45 de secunde. Puteți schimba pauza folosind comanda `TIMEOUT` astfel:

```
TIMEOUT 10 ogin: naba TIMEOUT 5 word: parola-mea
```

În acest caz, perioada de pauză este de 10 secunde când `chat` așteaptă promptul *login*:. Apoi pauza este schimbată la 5 secunde înainte ca `chat` să caute șirul *Password*:

Secvențe sub-expect

Secvența tipică de intrare în sistem ilustrează un caz simplu. Un caz mai complex este atunci când `chat` caută un prompt de conectare, dar acesta nu sosește în perioada de pauză. Dacă v-ați fi conectat la un terminal, ați fi apăsător `Enter` pentru a obține alt prompt de conectare. Puteți simula acest lucru utilizând secvențele sub-expect ale programului `chat`, care încep cu două liniuțe. Iată un exemplu:

```
ogin:--ogin: naba word: parola-mea
```

În acest caz, dacă `chat` nu primește promptul *login*: în perioada pauzei, `chat` trimite un singur retur (exact ca și cum ați fi apăsător `Enter` de la tastatură) și apoi caută din nou promptul *login*:

Șiruri ABORT (de renunțare)

Cele mai multe modemuri raportează starea unei conexiuni sub forma unui șir. De exemplu, modemul poate afișa `CONNECT` (care poate fi urmat de viteza conexiunii) atunci când conexiunea este realizată, `BUSY` dacă telefonul calculatorului de la distanță este ocupat și `NO CARRIER` dacă modemul nu primește ton de la telefon.

Veți dori probabil să opriți conexiunea dacă modemul afișează orice stare diferită de `CONNECT`. Puteți face acest lucru folosind comenzile `ABORT` ale programului `chat`. Ideea este să indicați șiruri care să determine programul `chat` să renunțe la legătura telefonică. Iată un exemplu:

```
ABORT BUSY ABORT "NO CARRIER" " " ATZ OK ATDT5551212
```

Prima pereche `ABORT BUSY` definește șirul `BUSY` ca un șir de renunțare. Următoarea pereche, `ABORT "NO CARRIER"`, definește alt șir de renunțare. (Folosiți ghilimelele pentru a încadra orice șir ce conține spații.)

După cele două șiruri de renunțare, apar două perechi expect-send (așteptare-trimitere). Prima pereche instruieste programul *chat* să aștepte caracterul vid (indicat printr-un șir vid, "") și să trimită comanda ATZ pentru a reseta modemul. Următoarea pereche expect-send determină chat să aștepte șirul OK și să trimită comanda de formare (ATDT, urmată de un număr de telefon) după ce modemul trimite șirul OK.

Secvențe escape

Puteți folosi caractere speciale în șirurile expect-send. Caracterele speciale cel mai frecvent folosite sunt indicate prin prefixul backslash (\), care servește drept caracter escape. Aceste secvențe de caractere speciale sunt numite în mod tradițional *secvențe escape*. Tabelul 18-3 listează secvențele escape pe care le puteți folosi în programele script pentru programul *chat*.

Tabelul 18-3 Secvențe escape pentru programul chat

Secvență	Descriere
" "	Două ghilimele consecutive indică un șir vid. Dacă trimiteți un șir vid, <i>chat</i> trimite un singur retur de car.
\b	Un caracter backspace.
\c	Oprește caracterul de linie nouă la capătul șirului trimis. Puteți trimite \c numai la capătul unui șir.
\d	Produce o întârziere de o secundă. Nu puteți folosi această secvență în șirul expect.
\ddd	Un bait ce conține valoarea specificată exprimată în baza 8.
\K	Trimite un semnal de întrerupere (BREAK) (vezi Capitolul 15). Nu puteți folosi această secvență într-un șir expect.
\n	Un caracter de salt la linie nouă.
\N	Trimite un caracter nul (un bait cu toți biții zero). Nu puteți folosi această secvență într-un șir expect.
\p	Face o pauză de o zecime de secundă. Nu puteți folosi această secvență într-un șir expect.
\q	Suprimă scrierea șirului în fișierul /var/adm/syslog. Această secvență nu este validă într-un șir expect.
\r	Un retur de car.
\s	Un caracter spațiu. Folosiți această secvență când nu doriți să încadrați șirul între ghilimele.
\t	Un tabulator.
^C	Un caracter de control corespunzător literei ce urmează după accentul circumflex. Astfel, ^X înseamnă Ctrl-X.

Puteți folosi toate aceste secvențe escape în șiruri send; dar, așa cum indică comentariile din tabel, sunt multe secvențe pe care nu le puteți folosi în șiruri expect.

Folosirea pppd cu un script chat

Detaliile importante ale rețelelor PPP sunt gestionate de programul daemon PPP, pppd, care folosește codul driverului PPP în nucleul Linux pentru a schimba pachete TCP/IP prin portul serial.

Pentru a realiza o conexiune PPP cu un calculator de la distanță trebuie să parcurgeți două etape de bază:

1. Formați numărul de telefon utilizând modemul și executați login pe calculatorul de la distanță, folosind un nume de utilizator și o parolă furnizate de către proprietarul acestuia. De obicei, când executați login pe sistemul de la distanță, această acțiune trebuie să lanseze în execuție programul PPP.

2. Porniți programul daemon PPP (pppd) de pe sistemul dumneavoastră pentru a iniția și dirija o sesiune PPP cu sistemul de la distanță.

Programul pppd este conceput pentru a îndeplini cele două operațiuni de mai sus. În primul rând, folosiți programul chat pentru a executa primul pas: formarea numărului de telefon și execuția de login pe calculatorul de la distanță. Nu este nevoie să rulați chat separat. Trebuie doar ca în linia de comandă a programului pppd să inserați un script chat; pppd va invoca chat pentru procesarea scriptului respectiv.

O linie de comandă tipică pppd

În cea mai simplă formă, puteți rula pppd cu o linie de comandă lungă care ar putea arăta astfel:

```
/usr/lib/ppp/pppd connect "/usr/lib/ppp/chat chat-script" crtscts modem \
defaultroute 192.168.101.111:192.168.102.1 /dev/cua0 38400
```

Caracterul backslash (\) de la sfârșitul primei linii este folosit pentru a continua comanda pe linia următoare. O linie de comandă pppd reală este chiar mai lungă, pentru că porțiunea cu scriptul chat este o secvență de perechi expect-send. În continuare, prezentăm o listă cu semnificațiile diferitelor părți dintr-o comandă pppd:

■ Opțiunea connect instruieste pppd să se conecteze executând comanda încadrată între ghilimele duble scrisă după opțiunea connect. În exemplu, comanda nu conține ghilimelele ce o încadrează:

```
/usr/lib/ppp/chat chat-script
```

■ Partea chat-script poate fi o secvență lungă de perechi de șiruri expect-send pe care programul chat le procesează. Puteți implementa un simplu script de conectare dial-up folosind următoarea comandă pentru a porni chat:

```
/usr/lib/ppp/chat „ ATZ OK ATDT5551212 CONNECT „ ogin: my-
username \
word: my - password
```

■ Acest script chat trimite ATZ pentru a reseta modemul; așteaptă un OK și apoi trimite o comandă de formare a numărului de telefon (ATDTnumar-de-telefon); așteaptă un CONNECT și apoi trimite un șir gol (un retur de car); așteaptă șirul ogin: și trimite my-username (care ar trebui să fie numele dumneavoastră de intrare în sistem PPP); în sfârșit, așteaptă șirul word: și trimite my-password (parola pentru contul dumneavoastră de conectare PPP).

■ Opțiunea crtscts instruieste pppd să folosească semnalele Request to Send (RTS) și Clear to Send (CTS) pentru dialogul de confirmare (handshaking) hardware cu modemul. (Consultați Capitolul 15, „Modemuri“, pentru informații suplimentare despre semnalele RTS și CTS.)

■ Opțiunea modem face ca pppd să folosească semnalele de control pentru modem ce reflectă starea acestuia.

■ Opțiunea defaultroute instruieste pppd să creeze o intrare pentru o rută prestabilită pe sistemul dumneavoastră pentru a folosi calculatorul de la distanță ca poartă de acces. Dacă aveți deja creată o rută prestabilită (pentru rețeaua dumneavoastră locală Ethernet, de exemplu), pppd nu creează ruta prestabilită pentru calculatorul de la distanță.

■ Cele două adrese IP instruiesc pppd să folosească adresa IP 192.168.101.111 pentru interfața locală și adresa IP 192.168.102.1 pentru capătul de la distanță al conexiunii PPP. Dacă calculatorul de la distanță atribuie adresele IP în mod dinamic, puteți folosi 0.0.0.0 ca adresă locală IP; pppd folosește automat adresa IP atribuită de calculatorul de la distanță.

■ Numele dispozitivului indică programului pppd să apeleze în exterior dispozitivul /dev/cua0 (COM1). Dacă modemul dumneavoastră este pe COM2, folosiți /dev/cua1 ca nume al dispozitivului.

■ Ultima opțiune determină pppd să stabilească rata de transfer a datelor pentru modem (numită deseori *viteză de baud*) la 38.400 bps.

Așa cum se vede în lista de mai sus, linia de comandă pppd conține o mulțime de informații. În același timp, execută o mulțime de operații. După ce pppd stabilește conexiunea cu calculatorul de la distanță, sistemul dumneavoastră devine parte a rețelei de la distanță (cu condiția ca sistemul de la distanță să adauge intrările corespunzătoare în tabela de rutare).

Ar trebui să rețineți un element cheie despre linia de comandă pppd: întreaga comandă chat (care urmează după opțiunea connect) este încadrată între ghilimele, dar comanda chat însăși folosește diferite șiruri încadrate între ghilimele. Nu uitați să folosiți prefixul backslash (\) pentru acele ghilimele înglobate. Exemplul următor prezintă aspectul unui script chat într-o linie de comandă pppd:

```
/usr/lib/ppp/pppd connect "/usr/lib/ppp/chat \"\" ATZ OK ATDT5551212
CONNECT \"\" \
ogin: my-username word: my-password" (restul liniei de comanda a fost omis).
```

Așa cum arată acest exemplu, toate ghilimelele înglobate necesită prefixul backslash (\). Dacă comanda AT pentru modem include semnul ampersand (&), mai aveți nevoie de prefixul backslash pentru fiecare ampersand. Linia de comandă *chat* o puteți încadra între ghilimele simple; în cazul acesta nu mai este necesar prefixul backslash (\) în fața ghilimelelor duble înglobate.

Un script PPP dial-up

Dacă aveți de gând să folosiți pppd frecvent, probabil că nu doriți să introduceți comanda de la tastatură. Ar trebui să pregătiți un script shell și să lansați pppd din acel script. Dacă salvați fișierul script cu numele dial-ppp, de exemplu, puteți seta o conexiune PPP cu următoarea comandă:

dial-ppp

Așa procedez eu când îmi conectez PC-ul Linux la Internet prin intermediul furnizorului de servicii Internet.

Iată un script shell tipic pentru setarea unei conexiuni PPP cu un calculator de la distanță:

```
#!/bin/sh
# Un script pentru stabilirea unei conexiuni PPP dial-up

# Variabila shell cu adresa IP a capatului de la distanta
RMTIP="192.168.102.1"

# Daca exista legatura PPP, atunci iese
/sbin/ifconfig | grep $RMTIP >/dev/null && exit 0

# Alte variabile shell
# Seteaza LOCIP la " " daca adresa dumneavoastra IP este atribuita dinamic
LOCIP="192.168.101.111"
PHONE="301-555-1212"
DEVICE="/dev/ttyS0"
SPEED="38400"
PPP_LOGINNAME="my-username"
PPP_PASSWORD="my-password"

/usr/sbin/pppd connect "/usr/sbin/chat -v REPORT CONNECT \
ABORT \"NO CARRIER\" ABORT BUSY ABORT \"NO DIAL TONE\" \
```

```
\"\" ATZ OK ATE1M1V1X4L3S0=0\&c1Q0DT$PHONE CONNECT \"\" \
ogin:--ogin: $PPP LOGINNAME word: $PPP_PASSWORD" \
rtscts modem defaultroute $LOCIP:$RMTIP $DEVICE
$SPEED
```

Sfarsitul fisierului script

Am înlocuit câteva informații importante (ca de exemplu numărul de telefon, numele utilizatorului, parola și adresele IP) cu valori arbitrare. În fișierul script această informație apare scrisă cu litere cursive. Trebuie să înlocuiți acești parametri cu informațiile dumneavoastră. Dacă aveți un cont PPP la un ISP (furnizor de servicii Internet), atunci acesta trebuie să vă furnizeze datele respective (numărul de telefon, adresele IP, numele utilizatorului și parola). Trebuie să editați, de asemenea, secvența de comandă AT pentru a putea lucra cu modemul dumneavoastră.

Testarea conexiunii PPP

Dacă aveți acces PPP la alt sistem (ca de exemplu al unui ISP sau cu un sistem de la locul dumneavoastră de muncă), puteți crea un script similar celui descris în secțiunea precedentă și să beneficiați de accesul total la alt sistem prin TCP/IP. După ce rulați fișierul script și după ce programul pppd încheie schimburile de protocol inițiale pentru crearea conexiunii, puteți verifica dacă conexiunea este stabilită invocând comanda ifconfig, care ar trebui să afișeze o listă similară cu cea prezentată în continuare:

```
/sbin/ifconfig
lo Link encap:Local Loopback
inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING MTU:3584 Metric:1
RX packets:1738 errors:0 dropped:0 overruns:0
TX packets:1738 errors:0 dropped:0 overruns:0

eth0 Link encap:Ethernet Hwaddr 00:20:AF:E8:A2:25
inet addr:192.168.1.1 Bcast:192.168.1.255
Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
RX packets:146900 errors:6 dropped:6 overruns:0
TX packets:120970 errors:0 dropped:0 overruns:0
Interrupt:10 Base address:0x210

ppp0 Link encap:Point-to-Point Protocol
inet addr:204.192.67.100 P-t-P:204.91.204.91
Mask:255.255.255.0
UP POINTOPOINT RUNNING MTU:1500 Metric:1
RX packets:492 errors:0 dropped:0 overruns:0
TX packets:682 errors:0 dropped:0 overruns:0
```

Ar trebui să găsiți dispozitivul ppp0 inclus între datele de ieșire. Mesajul emis de ifconfig prezintă, de asemenea, adresele IP pentru capătul local și cel de la distanță al conexiunii PPP. Acest mesaj confirmă că dispozitivul PPP există și funcționează.

Pentru a verifica dacă tabela de rutare este corect configurată, folosiți comanda /sbin/route fără argumente, astfel:

```
/sbin/route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default-gateway * 255.255.255.255 UH 0 0 0 ppp0
192.168.1.0 * 255.255.255.0 U 0 0 117 eth0
127.0.0.0 * 255.0.0.0 U 0 0 60 lo
default default-gateway 0.0.0.0 UG 0 0 2 ppp0
```


În tabela de rutare, prima linie prezintă o rută către capătul de la distanță al conexiunii PPP; acesta ar trebui setat pe dispozitivul ppp0. De asemenea, ruta prestabilită ar trebui setată astfel încât capătul de la distanță al conexiunii PPP să servească drept poartă de acces pentru sistemul dumneavoastră (așa cum o arată ultima linie din tabela de rutare).

După verificarea configurației interfeței (cu ajutorul comenzii `ifconfig`) și a tabelii de rutare (cu ajutorul comenzii `route`), ar trebui să verificați dacă puteți accesa câteva calculatoare gazdă binecunoscute. Dacă ISP-ul dumneavoastră v-a dat adresa IP a unui server de nume sau a unui server poștal, puteți încerca să contactați acele adrese. Altfel, încercați să trimiteți comanda `ping` către adresa IP a unui sistem de la locul dumneavoastră de muncă sau de la universitate.

Următorul exemplu vă indică un posibil rezultat al comenzii `ping`:

```
ping 140.90.23.100
PING (140.90.23.100): 56 data bytes
64 bytes from 140.90.23.100: icmp_seq= 0 ttl= 244 time=336.0 ms
64 bytes from 140.90.23.100: icmp_seq= 1 ttl= 244 time=290.2 ms
64 bytes from 140.90.23.100: icmp_seq= 2 ttl= 244 time=350.2 ms
64 bytes from 140.90.23.100: icmp_seq= 3 ttl= 244 time=300.2 ms
64 bytes from 140.90.23.100: icmp_seq= 4 ttl= 244 time=340.2 ms
64 bytes from 140.90.23.100: icmp_seq= 5 ttl= 244 time=290.2 ms
```

```
--- 140.90.23.100 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 290.2/317.8/350.2 ms
```

Sfârșitul fiecărei linii arată timpul dus – întors pentru un pachet ce se deplasează din sistemul dumneavoastră către adresa IP desemnată (140.90.23.100, în acest caz) și apoi se întoarce în sistemul dumneavoastră. Pentru o conexiune pe o linie dial-up, timpul va fi exprimat în sute de milisecunde.

Nu trebuie să aveți un cont pe un sistem pentru a putea invoca comanda `ping` cu adresa IP a acestuia. Cu toate că un sistem poate dezactiva răspunsul automat la comanda `ping` (`ping` folosește mesajele ICMP), cele mai multe sisteme emit un răspuns la această comandă.

Terminarea conexiunii PPP

După ce `pppd` stabilește o conexiune PPP, legătura se păstrează atât timp cât programul PPP de la ambele capete continuă să ruleze. Când nu mai aveți nevoie de conexiunea PPP, puteți să o întrerupeți oprind procesul `pppd`.

După crearea unei legături PPP, `pppd` creează un fișier în directorul `/var/run`, unde stochează identificatorul de proces pentru `pppd`. Numele fișierului are la bază numele dispozitivului PPP. Pentru prima legătură PPP dispozitivul PPP este `ppp0`, iar fișierul cu identificatorul procesului este `/var/run/ppp0.pid`. Puteți scrie un script shell pentru a distruge procesul `pppd` folosind identificatorul acestuia stocat în fișierul `/var/run/ppp0.pid`.



Secret

De fapt, nu este necesar să vă scrieți propriul script de interpretor pentru a termina o legătură PPP. În directorul `/usr/doc/ppp-2.2.0f/scripts/` veți găsi un fișier script numit `ppp-off` care execută această operație. Oricând doriți să întrerupeți legătura PPP, intrați în sistem ca `root` și tastați `/usr/doc/ppp-2.2.0f/scripts/ppp-off`.

Această comandă încheie procesul `pppd`, care la rândul său videază tabela de rutare.

Fișierul script `/usr/doc/ppp-2.2.0f/scripts/ppp-off` se prezintă astfel:

```
#!/bin/sh
```

```
DEVICE=ppp0
```

```
#
# Daca fisierul pid ppp0 este prezent, programul ruleaza. Opriti-l.
if [ -r /var/run/$DEVICE.pid ]; then
    kill -INT cat/var/run/$DEVICE.pid
#
# Daca operatia a esuat, asigurati-va ca fisierul pid este eliminat.
#
    if [ ! "$?" = "0" ]; then
        echo "removing stale $DEVICE pid file."
        rm -f /var/run/$DEVICE.pid
        exit 1
    fi
#
# Operatia a reusit. Terminare cu starea adecvata.
#
    echo "$DEVICE link terminated"
    exit 0
fi
#
# Legatura nu este activa
#
echo "$DEVICE link is not active"
exit 1
```



Observați că acest script folosește comanda `cat /var/run/$DEVICE.pid` (`DEVICE` este definit ca `ppp0`) pentru a obține identificatorul procesului `pppd`. Acel identificator de proces este folosit apoi ca argument în comanda `kill`, care distruge procesul `pppd`.

Rutarea prin conexiunea PPP

O întrebuintare uzuală a unei conexiuni PPP este de a conecta două rețele separate geografic, sau și mai uzual, pentru a conecta o mică rețea locală la Internet. Iată un scenariu tipic: aveți o mică rețea locală Ethernet pe care doriți s-o conectați la Internet. Puteți face acest lucru cu un PC Linux care are o placă Ethernet și un modem. Sistemul Linux este prezent în rețeaua dumneavoastră Ethernet prin interfața plăcii Ethernet. Dacă puteți stabili o conexiune PPP cu un sistem din Internet, puteți folosi PC-ul Linux ca poartă de acces între rețeaua dumneavoastră locală și calculatorul de la distanță (care, probabil, este deja conectat la Internet). Figura 18-2 ilustrează un astfel de scenariu.

În acest caz, aveți o mică rețea locală Ethernet cu câteva PC-uri care are adresa IP de clasa C 206.197.168.0. Ați atribuit adresa IP 206.197.168.200 plăcii Ethernet a PC-ului dumneavoastră Linux.

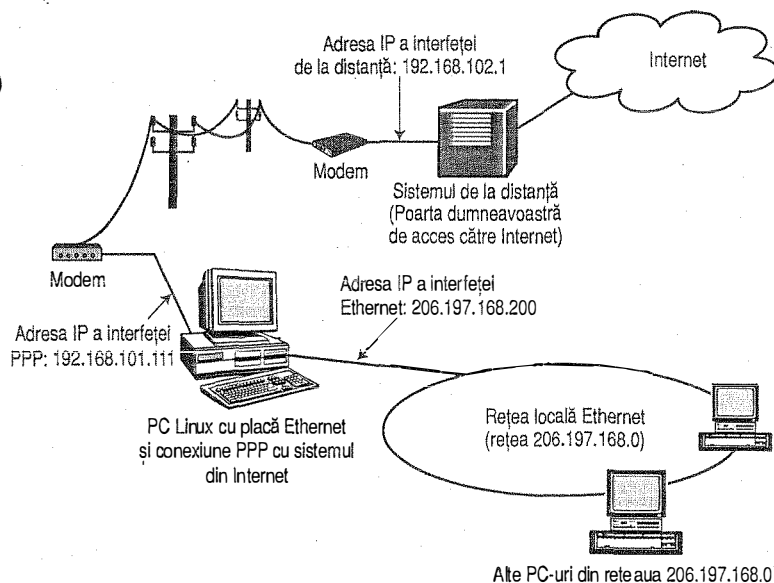
PC-ul Linux are, de asemenea, un modem prin care stabiliți o conexiune PPP cu un calculator de la distanță din Internet. Ambele capete ale conexiunii PPP au adrese IP unice. Doriți să vă asigurați că PC-urile din rețeaua dumneavoastră locală pot accesa Internetul. Doriți să folosiți PC-ul dumneavoastră Linux ca poartă de acces pentru a ruta pachetele de rețea între rețeaua Ethernet și calculatorul de la distanță, pentru că PC-ul Linux este singurul ce conține ambele interfețe, PPP și respectiv Ethernet. Puteți realiza acest lucru într-unul din următoarele moduri:

■ **Folosiți-vă PC-ul Linux ca router:** Obțineți o adresă de rețea de clasa C valabilă de la un ISP (furnizor de servicii Internet) și configurați PC-ul Linux astfel încât să funcționeze ca poartă de acces între rețeaua dumneavoastră și Internet.

■ **Folosiți mascarea IP:** Creați-vă rețeaua locală Ethernet cu adrese IP private (de exemplu, folosiți 192.168.1.0. ca adresă a rețelei) și folosiți PC-ul Linux ca parafo (firewall). Activați

deghizarea IP (IP masquerading) în nucleul Linux și folosiți programul `ipfwadm` pentru a permite PC-ului Linux să împrumute identitatea oricăruia din sistemele rețelei dumneavoastră. În esență, fiecare sistem din rețeaua dumneavoastră devine capabil de a accesa Internetul, chiar dacă nu are adresă IP unică.

Figura 18-2
Conectarea unei rețele locale (LAN) la Internet



Următoarele secțiuni explică, în continuare, cele două metode de mai sus.

Folosirea PC-ului Linux ca router

Pentru a folosi PC-ul dumneavoastră Linux ca router, trebuie să parcurgeți următorii pași:

1. Activați expedierea IP (IP forwarding) în nucleul Linux. Când reconfigurați nucleul folosind comanda `make config` (vezi capitolul 2), trebuie să răspundeți afirmativ la următoarele întrebări (printre alte întrebări legate de TCP/IP):

```
TCP/IP networking (CONFIG_INET) [Y/n/?] y
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?] y
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] y
IP: syn cookies (CONFIG_SYN_COOKIES) [Y/n/?] y
IP: accounting (CONFIG_IP_ACCT) [Y/n/?] y
IP: optimize as router not host (CONFIG_IP_ROUTER) [N/y/?] y
```

Când expedierea IP este activată, nucleul Linux trimite automat pachete de la o interfață la alta.

2. Folosiți comanda `route` pentru a stabili o rută către calculatorul de la distanță. O rută specifică o cale între două noduri de rețea. O specificație tipică arată cam așa: Toate pachetele pentru rețeaua 206.197.168.0 vor fi trimise la interfața Ethernet `eth0`. Adresa IP a porții de acces este folosită ca destinație pentru orice adresă de rețea care nu are ruta explicită. Dacă aveți o legătură PPP, ați putea spune „Trimiteți orice pachet cu adresa nedeterminată la adresa IP a capătului de la distanță al legăturii PPP.” Trebuie să folosiți adresa IP a capătului de la distanță al conexiunii PPP ca poartă de acces către sistemul Linux. Pentru a învăța mai multe despre comanda `route`, tastați `man route` la promptul interpretorului.

3. Asigurați-vă că toate sistemele din rețeaua dumneavoastră locală folosesc PC-ul Linux ca poartă de acces prestabilită pentru protocoalele de rețea TCP/IP.

4. Asigurați-vă că poarta de acces de la distanță are o rută către rețeaua dumneavoastră. Fără acest pas, pachetele TCP/IP din Internet nu pot ajunge la rețeaua dumneavoastră. Dacă ați obținut conexiunea la Internet de la un ISP (furnizor de servicii Internet), atunci acesta vă poate furniza (în schimbul unei taxe, bineînțeles) adresele IP corect direcționate pentru rețeaua dumneavoastră locală.



Observație

Cu toate că un PC Linux cu o conexiune PPP se poate comporta ca o poartă de acces între o rețea locală și Internet, o soluție mai economică ar putea fi folosirea unui router dial-up proiectat special pentru a conecta rețeaua dumneavoastră la Internet. În ambele cazuri, aveți nevoie de un furnizor de servicii Internet care să vă faciliteze accesul la un sistem de la distanță, conectat deja la Internet.

Folosirea deghizării IP

Folosiți deghizarea (masquerading) IP atunci când doriți să conectați o rețea locală Ethernet cu o adresă IP privată la Internet, prin intermediul unui PC Linux (care are atribuită o adresă IP în mod oficial) care are o conexiune la Internet.

Cu deghizarea IP activată, PC-ul dumneavoastră Linux se comportă ca o dublură pentru oricare dintre sistemele din rețeaua locală Ethernet. La fel ca și routerul, PC-ul Linux este desemnat ca poartă de acces pentru rețeaua Ethernet. În orice caz, deghizarea înseamnă mai mult decât simpla trimitere a pachetelor IP înainte și înapoi între rețeaua locală și Internet.

Când PC-ul Linux preia identitatea altui sistem din rețea, pachetele expediate se modifică astfel încât par întotdeauna că sunt emise de PC-ul Linux. Când se primește un răspuns la unul din pachetele expediate, PC-ul Linux realizează procesul invers – modifică pachetele astfel încât ele par să vină din Internet direct la sistemul care a trimis pachetul expediat. Rezultatul final este acela că fiecare sistem din rețeaua locală Internet, pare să aibă acces total la Internet chiar dacă rețeaua Ethernet folosește o adresă IP privată care nu este unică.

Pentru a activa și folosi deghizarea IP, trebuie să parcurgeți următorii pași:

1. Activați facilitățile IP forwarding, IP firewalling și IP masquerading în nucleul Linux. Pe măsură ce reconfigurați nucleul folosind comanda `make config` (vezi capitolul 2), trebuie să răspundeți afirmativ la următoarele întrebări (printre alte întrebări despre protocoalele de rețea TCP/IP):

```
*
* Opțiuni de lucru în rețea
*
Network firewalls (CONFIG_FIREWALL) [Y/n/?] y
Network aliasing (CONFIG_NET_ALIAS) [Y/n/?] y
TCP/IP networking (CONFIG_INET) [Y/n/?] y
IP: forwarding/gatewaying (CONFIG_IP_FORWARD) [Y/n/?] y
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] y
IP: syn cookies (CONFIG_SYN_COOKIES) [Y/n/?] y
IP: rst cookies (CONFIG_RST_COOKIES) [N/y/?]
IP: firewalling (CONFIG_IP_FIREWALL) [Y/n/?] y
IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE) [Y/n/?] y
IP: masquerading (CONFIG_IP_MASQUERADE) [Y/n/?] y
*
* Suportul de mascare specific protocoalelor va fi construit ca module.
*
IP: ipautofw masq support (CONFIG_IP_MASQUERADE_IPAUTOFW) [Y/n/?] y
```

```
IP: ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) [Y/n/?] y
IP: transparent proxy support (EXPERIMENTAL)
(CONFIG_IP_TRANSPARENT_PROXY) [Y/n/?] y
IP: always defragment (CONFIG_IP_ALWAIIS_DEFRAG) [Y/n/?] y
```

2. Asigurați-vă că PC-ul Linux are o conexiune la Internet și o conexiune cu rețeaua dumneavoastră locală. De obicei, PC-ul Linux are două interfețe de rețea – o placă Ethernet pentru rețeaua locală (LAN) și o conexiune PPP de tip dial-up pentru Internet (printr-un ISP).
3. Asigurați-vă că toate sistemele din rețea folosesc PC-ul Linux ca poartă de acces prestabilită pentru protocoalele de rețea TCP/IP.
4. Rulați `ipfwadm` – programul de administrare pentru parafofurile IP – pentru a seta regulile ce permit PC-ului Linux să se împrumute identitățile tuturor sistemelor din rețeaua dumneavoastră locală. De exemplu, pentru a activa deghizarea pentru o rețea locală a cărei adresă de clasa C este 192.168.1.0, puteți folosi următoarele comenzi:

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

Dacă doriți ca deghizarea IP să fie creată la inițializarea sistemului, trebuie să introduceți aceste comenzi în fișierul `/etc/rc.d/rc.local`.

Pentru a învăța mai multe despre programul `ipfwadm`, tastați `man ipfwadm`. Ar trebui, de asemenea, să citiți fișierul `IP-Masquerade mini-HOWTO`, descris la etapa 6.

5. Încărcați modulele care furnizează facilitățile de deghizare pentru anumite servicii Internet ca de exemplu File Transfer Protocol (FTP), Internet Relay Chat (IRC) și Real Audio. Ar trebui să adăugați următoarele linii la fișierul `/etc/rc.d/rc.local`:

```
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_radio
```

6. Pentru mai multe detalii despre deghizarea IP și `ipfwadm`, citiți fișierul `IP-Masquerade mini-HOWTO`. Mai întâi, tastați `cd /usr/doc/HOWTO/mini` pentru a schimba directorul. Apoi tastați `more IP-Masquerade` pentru a citi documentul. Puteți, de asemenea, consulta cea mai recentă versiune a acestui document pe Internet, la adresa <http://www.wvonline.com/~achau/ipmasq/>.



Sugestie

Veți considera deghizarea IP un mod convenabil de a furniza acces la Internet pentru o mică rețea locală (de exemplu la domiciliu sau la birou). Acasă eu folosesc deghizarea IP pentru a conecta o rețea locală Ethernet la Internet printr-un sistem Linux. PC-ul Linux are o placă Ethernet pentru conexiunea cu rețeaua locală și un modem pentru conectarea la Internet printr-un ISP (furnizor de servicii Internet). (Eu am o adresă IP validă de la ISP.) Cu deghizarea IP pe PC-ul Linux, fiecare membru al familiei poate accesa acum Internetul de pe oricare calculator al rețelei.

Configurarea unui server PPP

Secțiunile precedente prezintă modul în care PC-ul dumneavoastră Linux poate stabili o legătură PPP cu alt calculator care oferă servicii PPP. După ce o legătură PPP este creată, ambele capete ale acesteia au drepturi egale. Înainte ca o legătură PPP să fie stabilită, puteți considera capătul care inițiază conexiunea dial-up drept client, pentru că acel sistem are nevoie de conexiune. Celălalt capăt furnizează conexiunea PPP atunci când este nevoie, deci este serverul PPP.

Dacă vreți să permiteți altor utilizatori să se conecteze la PC-ul dumneavoastră Linux folosind PPP cu un modem dial-up, parcurgeți următorii pași:

1. Urmăriți etapele descrise în capitolul 15 pentru a activa login-ul de tip dial-up în sistemul dumneavoastră. Testați această parte pentru a fi siguri că totul funcționează; formați numărul de telefon al sistemului dumneavoastră din alt calculator și intrați în sistem ca utilizator. Acest pas presupune adăugarea unei linii în fișierul `/etc/inittab` similare cu următoarea:

```
s0:235:respawn:/sbin/uugetty ttyS0 F38400 vt100
```

2. Adăugați un nou utilizator pentru conexiunea PPP de tip dial-up – numele utilizatorului sub care sistemul client execută login. La login, programul `pppd` ar trebui să ruleze automat. Specificați `/usr/local/bin/ppp-login` ca program shell pentru utilizator.

3. Pregătiți fișierul script `/usr/local/bin/ppp-login`. Acest script este interpretorul de comenzi pentru contul de conectare PPP, ceea ce înseamnă că fișierul script este executat atunci când utilizatorul PPP se conectează. Astfel, scriptul ar trebui să lanseze programul `pppd` cu opțiunile corespunzătoare. După ce pregătiți fișierul script, nu uitați să-l faceți executabil. Următorul este un fișier script tipic `/usr/local/bin/ppp-login`:

```
#!/bin/sh
# Un script pentru pornirea serviciilor PPP pentru un cont de login
```

```
# Variabilele shell cu adrese IP
RMTIP= "192.168.111.111 "
LOCIP= "192.168.111.222 "
```

```
exec/usr/bin/pppd -detach silent modem \
crtsets $LOCIP:$RMTIP
```

Am testat această setare pe un PC Windows 95 și a funcționat; cu toate acestea, a trebuit să tastez numele utilizatorului și parola manual, într-o fereastră terminal. În rețelele cu conexiune dial-up a PC-urilor Windows 95, puteți activa o opțiune ce afișează o fereastră după ce modemurile stabilesc conexiunea. Aceasta este fereastra în care am tastat numele utilizatorului PPP și parola.

Rezumat

Mulți dintre noi nu au o conexiune directă la coloanele vertebrale cu lărgime mare de bandă ale Internetului. În schimb, ne bazăm pe conexiuni dial-up prin modem cu furnizorii de servicii Internet, care la rândul lor sunt conectați la magistrale (furnizate de obicei de mari companii de telecomunicații, ca de exemplu Sprint și MCI). Rețelele TCP/IP pe linii seriale pot fi realizate cu ajutorul protocoalelor SLIP și PPP. Linux este dotat cu software pentru SLIP și PPP. Acest capitol explică ce sunt SLIP și PPP și vă arată cum să configurați și să folosiți aceste două protocoale ca mecanism de transport al datelor pentru TCP/IP.

Din lectura acestui capitol ați învățat următoarele:

- Comunicațiile seriale cu programe precum `minicom` (sau ceva de genul `ProComm`, în lumea MS-DOS) permit PC-ului dumneavoastră să devină un terminal pentru un calculator de la distanță. Acest tip de comunicație serială nu vă oferă o conexiune de rețea completă, în care aceeași legătură fizică poate fi folosită de numeroase aplicații de rețea pentru transferul datelor între PC-ul dumneavoastră și cel de la distanță.

- ▶ SLIP sau Serial Line Internet Protocol a fost prima încercare de a folosi legătura serială ca nivel de transport al datelor pentru TCP/IP. Cu toate că este simplu, SLIP este deficitar în multe privințe. De exemplu, ambele capete necesită cunoașterea adreselor IP și numai un singur tip de protocol poate folosi o conexiune SLIP pentru transportul pachetelor.
- ▶ PPP sau Point-to-Point Protocol este un protocol mai complex, destinat transportului pachetelor în orice legătură punct la punct. PPP poate transporta pachete ale mai multor protocoale prin aceeași legătură. PPP este modalitatea preferată de a stabili o conexiune de rețea TCP/IP de tip dial-up cu un furnizor de servicii Internet.
- ▶ Un program numit `dip` este folosit pentru stabilirea unei conexiuni SLIP cu un calculator de la distanță.
- ▶ Crearea unei legături PPP presupune două etape elementare. Mai întâi trebuie să folosiți un program utilitar pentru a forma numărul și a face conexiunea cu modemul de la distanță, apoi trebuie să lansați programul PPP pe sistemul dumneavoastră. Pentru efectuarea celor două etape, se folosește programul `chat`, coroborat cu programul `pppd`.
- ▶ Când vă conectați PC-ul Linux la un calculator de la distanță folosind PPP, legătura PPP este folosită ca rută prestabilită pe PC-ul Linux, deoarece conexiunea PPP devine poartă de acces către restul Internetului.
- ▶ Puteți folosi deghizarea IP pentru a furniza acces la Internet unei rețele locale private printr-un PC Linux care are o conexiune PPP cu un ISP (furnizor de servicii Internet).
- ▶ De asemenea, vă puteți configura PC-ul Linux ca server PPP, astfel încât alți utilizatori să vă formeze numărul de telefon și să stabilească o conexiune PPP cu sistemul dumneavoastră. Acest capitol descrie etapele de configurare a unui server PPP.

Capitolul 19

Configurarea unui calculator gazdă Linux pentru Internet

În acest capitol

- ▶ Veți înțelege noțiunea de calculator gazdă din Internet
- ▶ Veți examina câteva servicii Internet
- ▶ Configurarea și testarea poștei
- ▶ Configurarea și folosirea mesajelor
- ▶ Afișarea mesajelor și verificarea distribuirii acestora
- ▶ Setarea unui server `ftp` anonim sigur

Experții văd Internetul ca un fundament al unui nou val al informaticii, în care informația este distribuită și afacerile sunt conduse prin intermediul rețelelor. Browserele Web, precum Netscape Navigator sau Microsoft Internet Explorer, oferă o imagine a noii lumi a informaticii, una în care puteți accesa informații din orice locație geografică la fel de ușor ca cele de pe discul dumneavoastră.

Sunteți probabil deja familiarizat cu poșta electronică (sau e-mail) și știrile (news) – facilitățile principale ale Internetului. Sistemul e-mail vă facilitează schimbul de mesaje și documente cu orice persoană din Internet. Grupurile de discuții (newsgroups) furnizează un sistem de buletine electronice (BBS) disponibil oriunde în lume.

Dacă doriți să profitați de ofertele Internet, trebuie să aveți acces la acesta. Dacă administrați o afacere sau dacă sunteți un împătimit al sistemului Linux, veți aprecia conectarea la Internet a PC-ului dumneavoastră Linux; setați-l ca un calculator gazdă în Internet.



Referință
încrucișată

Folosesc termenul de calculator gazdă în Internet (Internet host) pentru un sistem care are acces de rețea la Internet, ceea ce înseamnă că poate schimba pachete de rețea TCP/IP cu orice alt calculator din Internet. De obicei, sistemul dumneavoastră Linux stabilește o conexiune de rețea la Internet printr-o linie dial-up folosind SLIP și PPP. Capitolul 18 descrie protocoalele SLIP și PPP și vă arată, de asemenea, cum să vă conectați PC-ul Linux la Internet.

Pentru a fi un calculator gazdă în Internet, PC-ul dumneavoastră Linux are nevoie de mai mult decât conexiunea la Internet. Trebuie să setați un e-mail, FTP și mesaje. Dacă doriți să furnizați informații altor utilizatori, trebuie să vă creați propriul server Web. În plus, ar trebui să aveți grijă de securitatea datelor. În cele din urmă, sistemul dumneavoastră va fi pe Internet și oricine are acces la Internet poate (teoretic, cel puțin) să ajungă în sistemul dumneavoastră.

Acest capitol se concentrează asupra sarcinilor pe care trebuie să le realizați în mod obișnuit pentru a seta PC-ul dumneavoastră Linux ca un calculator gazdă pentru Internet. Învățați cum să setați sistemul e-mail, FTP-uri anonime (ce permit oricui să copieze fișiere din

sistemul dumneavoastră, dar într-o manieră destul de sigură) și grupuri de discuții. Acest capitol descrie, de asemenea, câteva modalități pentru protecția datelor din calculatorul dumneavoastră gazdă pentru Internet.



Observație

Setarea unui server Web este un proces detaliat descris în capitolul următor.

Ce este un calculator gazdă pentru Internet?

Înțelesul termenului de calculator gazdă pentru Internet (Internet host) depinde de ceea ce credeți că înseamnă *Internet*. Tehnic vorbind, Internetul este o rețea de rețele ce acoperă întreaga lume. Termenul *internet* (scris cu literă mică) este prescurtarea de la *internetworking* – adică interconectarea rețelelor. Protocolul Internet (IP) a fost proiectat pentru a conecta rețele. Așa cum ați învățat în capitolul 16, chiar și adresele IP au o porțiune alocată rețelei și una calculatorului gazdă.



Înălțor

Fizic, Internetul este similar cu o rețea de autostrăzi și drumuri. Această asemănare i-a determinat pe oameni să denumească Internetul „Superautostrada Informațională” (Information Superhighway). Exact așa cum rețeaua de autostrăzi și drumuri include câteva autostrăzi interstatale, multe drumuri statale și foarte multe străzi rezidențiale, Internetul are câteva rețele cu lățime de bandă foarte mare (magistrale T3 la 45 Mbps) și un număr mare de rețele de mică capacitate (cuprinzând de la conexiuni dial-up cu 28.800 bps la legături T1 cu 1,54 Mbps). Rețeaua cu lățime de bandă mare constituie coloana vertebrală a Internetului.

Spre deosebire de rețelele comerciale, ca de exemplu CompuServe și America Online, Internetul nu este condus de o singură organizație și nici nu este administrat de un calculator central. Vă puteți imagina rețeaua fizică Internet ca o „rețea a rețelelor” condusă colectiv de sute de organizații ce cooperează între ele. Îmi dau seama că o colecție de rețele condusă de sute de organizații sună incredibil, dar merge!

Din punct de vedere al conectivității fizice, un calculator gazdă din Internet este orice calculator dintr-o rețea care face parte din Internet. Totuși, utilizatorul judecă un calculator gazdă din Internet din punctul de vedere al serviciilor pe care acesta le furnizează. De obicei, utilizatorii se așteaptă la următoarele servicii uzuale de la un calculator gazdă pentru Internet:

- **Poștă electronică (e-mail).** Dintr-un calculator gazdă Internet, puteți trimite e-mail oricărui utilizator al rețelei Internet, folosind adrese ca de exemplu `president@whitehouse.gov`.
- **Grupuri de discuții (Newsgroups).** Puteți citi grupurile de discuții și să le difuzați știri acestora cu nume ca de exemplu `comp.os.linux.networking` sau `comp.os.linux.setup`.
- **Încărcarea informațiilor.** Puteți căuta informații cu utilitare precum *gopher* și browsere World Wide Web. Puteți, de asemenea, descărca fișiere folosind FTP. Reciproc, utilizatori de pe alte sisteme pot copia fișiere din sistemul dumneavoastră în mod obișnuit printr-o facilități numită *anonymous FTP* (FTP anonim). Veți învăța cum să setați serviciile anonime FTP în secțiunea „FTP anonim sigur” din acest capitol.
- **Acces la distanță.** Puteți folosi *telnet* pentru a vă conecta la alt calculator (calculatorul de la distanță) din Internet, presupunând că aveți acces la acel calculator.



Observație

Câteva facilități, ca de exemplu accesul la un calculator de la distanță cu *telnet* sau descărcarea fișierelor cu *ftp*, sunt active atunci când PC-ul dumneavoastră Linux este conectat la Internet. Pentru e-mail, grupuri de discuții și FTP anonim trebuie să parcurgeți câteva etape în Linux. Secțiunile următoare ale acestui capitol descriu configurarea acestor servicii Internet.

Schimbul de mesaje prin e-mail

Una din facilitățile cele mai folosite de utilizatorii Internetului este de a păstra legătura cu colegii, cu prietenii, cu cei dragi și chiar cu străini prin e-mail. Dacă n-ați utilizat prea mult sistemul e-mail, vă puteți întreba care-i marele avantaj. Chiar dacă ați folosit sistemul de e-mail intern al firmei la care lucrați, poate că nu veți aprecia facilitățile poștei electronice din Internet până când nu le încercați.

Puteți fi surprins de faptul că puteți trimite un mesaj unui prieten aflat la mii de kilometri depărtare și că puteți primi răspuns doar în câteva minute. Esențial este că puteți trimite mesaje oriunde în lume de pe un calculator gazdă din Internet și că aceste mesaje ajung la destinație în câteva minute, lucru ce nu poate fi realizat prin poșta clasică.

Deoarece sistemul e-mail poate fi stocat și trimis, puteți aranja să trimiteți și să primiți e-mail fără ca PC-ul dumneavoastră Linux să fie permanent calculator gazdă în Internet. Totuși, nu veți beneficia de livrarea aproape imediată a mesajelor dacă PC-ul dumneavoastră nu este calculator gazdă în Internet.

Participarea la grupuri de discuții (newsgroups)

Internetul vă ajută să comunicați în multe feluri. Prin e-mail puteți schimba mesaje cu oameni pe care-i cunoașteți deja. Uneori doriți să participați în grupuri de discuții. În cazul în care aveți nevoie de ajutor pentru setarea XFree86 pe o placă grafică ATI Mach64, de exemplu, doriți să consultați pe oricine are cunoștințe despre acest subiect. Pentru acel tip de comunicație, puteți lăsa un mesaj în grupul de discuții corespunzător; este probabil ca cineva să vă dea un răspuns.

Grupurile de discuții din Internet sunt ca buletinele electronice sau forumurile de pe alte servicii online, ca de exemplu CompuServe sau America Online. Veți găsi o mare varietate de grupuri de discuții, ce acoperă subiecte de la politică la calculatoare. Vă puteți gândi la grupurile de discuții din Internet ca la o întrunire – un loc de întâlnire virtual unde puteți pune întrebări și discuta diverse chestiuni.

Localizarea și parcurgerea informației

Probabil că ați încercat deja partajarea fișierelor între calculatoarele unei rețele locale (LAN). De obicei, o rețea locală are un server și orice utilizator din rețea poate accesa și folosi informații din fișierele depozitate pe server.

Fiind o colecție de rețele interconectate, Internetul permite gazdelor – calculatoarelor din Internet – să partajeze informații folosind o varietate de protocoale. De exemplu, FTP (File Transfer Protocol) arată cum să selectați și să copiați fișiere din alt calculator din Internet. Ați văzut câteva exemple de FTP în capitolul 2, unde am explicat cum să descărcați corecții (patches) pentru nucleul Linux.



Referință încrucigată

Alt protocol de partajare a informațiilor, mai recent, este Hypertext Transfer Protocol (HTTP), care permite calculatoarelor să facă schimb de documente formate în limbajul HTML (Hypertext Markup Language). HTML și HTTP reprezintă fundamentul pentru World Wide Web, prin intermediul căruia puteți

examina pur și simplu la documentele de pe alte calculatoare din Internet. Veți învăța mai multe despre World Wide Web în capitolul 20.

Utilizarea unei strategii simple pentru poștă și mesaje

Cu toate că folosirea poștei și a mesajelor este destul de simplă, ambele necesită efort de setare și întreținere corespunzătoare din partea dumneavoastră. Grupurile de discuții, în special, necesită mult spațiu pe disc și acces la un server de mesaje. Vă trebuie mult spațiu pe disc, pentru că există multe grupuri de discuții iar volumul mesajelor poate fi destul de mare. De asemenea, mesajele trebuie șterse periodic; altfel, discul se va umple. Chiar dacă alegeți un mic subset de grupuri de discuții, puteți avea nevoie de sute de megabaiți pentru a păstra mesajele pe sistemul dumneavoastră.



Sugestie

Dacă aveți acces la Internet printr-un ISP (Internet Service Provider), acesta vă furnizează accesul la un server poștal și un server de mesaje. Apoi, puteți folosi software pentru poștă care descarcă mesajele din serverul dumneavoastră poștal și un cititor de mesaje care citește grupurile de discuții direct de pe serverul de mesaje. Dacă sunteți singurul utilizator al PC-lui dumneavoastră Linux, vă recomand cu insistență această strategie pentru citirea mesajelor.



Referință încrucișată

Așa cum veți învăța în capitolul 20, puteți folosi un browser Web, ca de exemplu Netscape, pentru a citi mesaje și trimite corespondență, precum și pentru a accesa pagini Web.

Instalarea aplicațiilor software pentru poștă și mesaje

În timpul instalării sistemului Linux Red Hat de pe CD-ROM -ul anexat acestei cărți, puteți opta pentru instalarea pachetelor necesare pentru poștă și mesaje. La fel cum este descris și în capitolul 1, selectați componenta *Mail/WWW/News Tools* atunci când vi se solicită să indicați componentele pe care le doriți instalate.



Referință încrucișată

Dacă instalați softul de poștă și mesaje o dată cu Linux, nu mai aveți prea multe de făcut pentru a putea utiliza serviciile poștale și de mesaje. În caz contrar, puteți folosi utilitarul Red Hat Package Manager (RPM) pentru a instala pachete individuale. Capitolul 2 descrie modul de folosire a programului rpm pentru a instala aplicații software noi.

Pentru a accesa fișierele RPM pentru poștă și mesaje, inserați CD-ROM-ul în unitatea CD și tastați următoarea comandă:

```
mount /dev/cdrom /mnt/cdrom
```

Apoi intrați în directorul `/mnt/cdrom/RedHat/RPMS`. Veți găsi fișierele RPM localizate în acest director. Câteva din programele RPM pe care le puteți instala sunt următoarele (Eu vă voi indica numele RPM; numele complet al fișierului începe cu numele RPM, conține numărul versiunii și se termină cu extensia `.rpm`):

- `anonftp` (setarea protocolului FTP anonim)
- `elm` (un agent utilizator al poștei)
- `fetchmail` (un utilitar de regăsire și trimitere a poștei)

- `ftp` (client FTP)
- `imap` (server pentru Internet Message Access Protocol)
- `inews` (program Internet pentru afișarea mesajelor)
- `inn` (InterNetNews, un server de mesaje ce se bazează pe TCP/IP)
- `mailx` (un agent utilizator al poștei)
- `metamail` (software pentru gestionarea MIME-Multipurpose Internet Mail Extensions)
- `pine` (un program multifuncțional pentru poștă și mesaje)
- `procmail` (un pachet pentru livrarea locală a poștei)
- `sendmail` (un agent complex de transport al poștei)
- `sendmail -cf` (fișiere de configurare pentru sendmail)
- `sendmail -doc` (documentație pentru sendmail)
- `tin` (un cititor de mesaje orientat pe pagină)
- `trn` (un cititor de mesaje pentru un server de mesaje de la distanță)
- `uucp` (protocol de copiere UNIX-la-UNIX)

Probabil că aveți deja instalate multe dintre aceste pachete. Pentru acest capitol aveți nevoie numai de pachetele de poștă și mesaje, deci ar trebui să instalați `elm`, `sendmail`, `sendmail -cf`, `tin` și `trn` (precum și orice alt cititor de poștă sau mesaje pe care doriți să-l încercați).

Poșta

Poșta electronică – e-mail – este unul dintre cele mai populare servicii ale Internetului. Oricui îi place să trimită un mesaj fără să fie necesar să-și lase un altuia mesaje telefonice fără să între în contact. Când trimiteți un mesaj e-mail, acesta așteaptă în cutia poștală a destinatarului până când va fi citit.

Sistemul e-mail a început ca un mecanism simplu în care mesajele erau copiate într-un fișier de poștă al unui utilizator. Acel mecanism simplu mai este încă folosit. În Linux, mesajele dumneavoastră poștale sunt stocate în directorul `/var/spool/mail`, într-un fișier text al cărui nume este identic cu numele dumneavoastră de utilizator.

Mesajele sunt adresate unui nume de utilizator, ceea ce înseamnă că dacă John Doe se conectează cu numele `jdoe`, mesajul e-mail pentru el este adresat către `jdoe`. Singura informație de care mai aveți nevoie pentru a identifica în mod unic destinatarul este numele întreg al domeniului din care face parte sistemul destinatarului. Astfel, dacă sistemul lui John Doe se numește `someplace.net`, adresa lui completă de e-mail devine `jdoe@someplace.net`. Având această adresă, oricine de pe Internet îi poate trimite mesaje lui John Doe.



CD

Sistemul Linux aflat pe CD-ROM-ul anexat acestei cărți conține softul de care aveți nevoie pentru a seta și folosi sistemul e-mail pe sistemul dumneavoastră Linux. Următoarele secțiuni vă ghidează prin diferitele aspecte ale setării și folosirii sistemului e-mail pe sistemul dumneavoastră Linux.

Software pentru poștă

Pentru a seta și folosi e-mail pe PC-ul dumneavoastră Linux aveți nevoie de două tipuri de software pentru poștă:

- *Mail user agent* (agent utilizator al poștei) vă permite să vă citiți mesajele poștale, să scrieți răspunsuri și să compuneți mesaje noi.

■ *Mail transport agent* (agent ce transportă poșta) trimite și primește mesaje poștale sub formă de text. Metoda exactă folosită pentru transportul poștei depinde de rețea. În rețelele TCP/IP, agentul care transportă poșta o livrează folosind protocolul Simple Mail Transfer Protocol (SMTP). CD-ROM-ul anexat acestei cărți conține *sendmail*, un agent de transport al poștei puternic și popular pentru rețele TCP/IP.

Cei mai mulți dintre agenții ce transportă poșta rulează ca daemoni – procese ce rulează în fundal atât timp cât sistemul dumneavoastră este deschis. Deoarece dumneavoastră sau alt utilizator de pe sistem puteți trimite poșta oricând, agentul trebuie să fie acolo pentru a o livra la destinație. Agentul ce utilizează poșta rulează numai atunci când utilizatorul dorește să verifice poșta.



De obicei, un agent ce transportă poșta este activat după ce sistemul pornește. Fișierele de inițializare pentru Linux Red Hat sunt setate astfel încât agentul ce transportă poșta *sendmail* pornește când sistemul Linux este în modul multiutilizator. În fișierul `/etc/rc.d/init.d/sendmail`, veți găsi rândurile care lansează în execuție programul *sendmail*.

Mai multe despre *sendmail*

Acest capitol vă arată cum să folosiți un fișier de configurare predefinit *sendmail* pentru a putea primi mesaje e-mail. În orice caz, *sendmail* este un sistem de poștă foarte complex. Lucrarea *sendmail*, scrisă de Bryan Costales, Eric Allman și Neil Rickert (O'Reilly & Associates, 1993) vă va ajuta să configurați *sendmail*.

Deoarece sistemul este deja setat pentru a porni *sendmail* la inițializare, ceea ce mai aveți de făcut este să utilizați un fișier de configurare *sendmail* corespunzător pentru a putea primi e-mail pe sistemul dumneavoastră.

Fișierul de configurare pentru *sendmail*

Nu puteți trimite sau primi e-mail până când agentul de transport al poștei *sendmail* nu este configurat corespunzător. *sendmail* are reputația unui sistem de livrare a poștei complex și complet. Dacă aruncați o privire asupra fișierului de configurare pentru *sendmail*, `/etc/sendmail.cf`, veți fi imediat de acord că *sendmail* este într-adevăr complex. Din fericire, nu trebuie să deveniți expert în fișiere de configurare pentru *sendmail* (o carte întreagă a fost scrisă despre acest subiect; citiți paragraful „Mai multe despre *sendmail*”). Aveți nevoie doar de unul din fișierele de configurare predefinite conținute pe CD-ROM-ul anexat acestei cărți).

Dacă ați optat pentru componenta *Mail/WWW/News Tools* în timpul instalării Linux Red Hat (vezi capitolul 1), sistemul dumneavoastră ar trebui să aibă deja un fișier de configurare pentru *sendmail* care să funcționeze – `/etc/sendmail.cf`. Fișierul prestabilit atribuie o conexiune Internet și un server de nume. Dacă aveți o conexiune Internet, ar trebui să puteți primi și trimite e-mail de pe PC-ul dumneavoastră Linux.



Secret

Pentru a fi siguri că livrarea poștei funcționează corect, trebuie să vă asigurați că numele sistemului dumneavoastră se potrivește cu ceea ce v-a atribuit furnizorul dumneavoastră de servicii Internet. Cu toate că puteți atribui sistemului orice nume de calculator gazdă doriți, alte sisteme vă pot trimite poșta cu succes numai dacă numele sistemului dumneavoastră se află în serverul de nume al ISP-ului dumneavoastră.

Un test de livrare a poștei

Pe calculatorul meu, pot trimite și primi poștă imediat după selectarea unui fișier de configurare pentru *sendmail*. Mai întâi, m-am asigurat că numele calculatorului meu gazdă s-a potrivit cu ceea ce mi-a atribuit ISP-ul pentru utilizare. Apoi am folosit comanda *mail* pentru a compune și trimite un mesaj poștal mie însumi la o adresă diferită, după cum urmează:

```
mail nab@aaccess.digex.net
Subject: Testing e-mail
This is from my Linux PC.
```

EOT

Comanda *mail* activează un simplu agent utilizator de poștă. În acest exemplu, eu specific destinatarul – *nab@aaccess.digex.net* – în linia de comandă. Programul de poștă solicită introducerea subiectului care face obiectul mesajului. După subiect introduc mesajul și îl închei cu un rând care conține numai un punct (.). După ce termin mesajul, agentul utilizator de poștă predă mesajul programului *sendmail* – agentul ce transportă poșta – pentru a-l livra la adresa specificată. Deoarece sistemul meu era deja conectat la Internet, *sendmail* a livrat mesajul poștal imediat.

Pentru a verifica livrarea poștei, am folosit *telnet* pentru a mă conecta la *access.digex.net* și apoi mi-am verificat poșta, după cum urmează:

```
access2% mail
Mail version SMI 4.0 Fri Oct 14 12:50:06 PDT 1994 Type ? for help.
"/usr/spool/mail/naba": 1 message 1 new
>N 1 nab@adcc05211.slip.digex.net Sat May 16 17:17 16/707
Testing e-mail
Mail> (apasati Enter)
Message 1:
From nab@adcc05211.slip.digex.net Sat May 16 17:17:41 1998
Date: Sun, 17 May 1998 04:22:16 -0400
From: Naba Barkakati <nab@adcc05211.slip.digex.net>
To: nab@aaccess.digex.net
Subject: Testing e-mail
```

This is from my Linux PC.

```
Mail>r
To: nab@adcc05211.slip.digex.net
Subject: Re: Testing e-mail
```

```
Received your message.
Cc:
Mail>q
Saved 1 message în /homea/naba/mbox
access2%
```

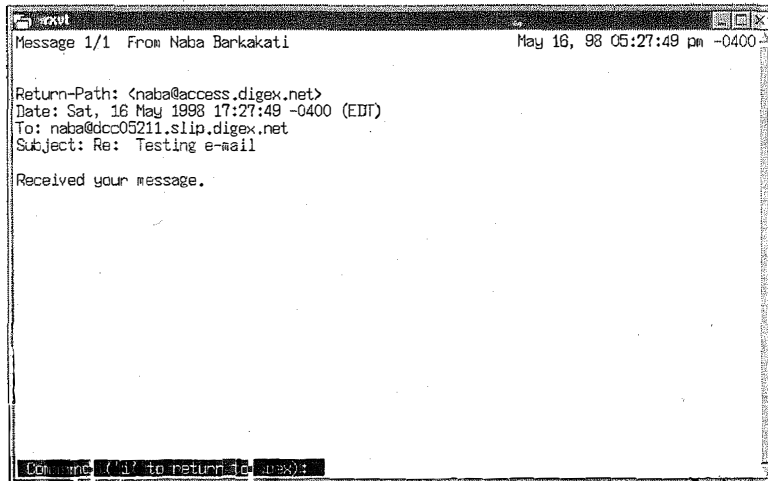
După cum puteți verifica din acest listing, mesajul meu poștal a ajuns la destinație. După citirea lui, am tastat *r* pentru a începe formularea răspunsului. Am scris un scurt răspuns și am apăsat *Ctrl-D* (astfel programul pentru poștă din *access.digex.net* înțelege că mesajul s-a încheiat). Programul pentru poștă solicită o listă pentru a face copii (adrese ce ar trebui să primească o copie a răspunsului) ce trebuie introdusă după textul *Cc:*. Am tastat *Enter* pentru a indica faptul că nu doresc să fie trimisă nici o copie la nici o adresă. La prompt-ul afișat am tastat *q* pentru a ieși din program.

După ce m-am deconectat de la *access.digex.net*, am verificat poșta din nou pe PC-ul meu Linux pentru a vedea dacă răspunsurile mele s-au întors înapoi. Pentru a citi mesajele poștale am utilizat programul *e1m*. După ce am tastat *e1m*, a apărut o fereastră în care erau afișate

mesajele curente din cutia mea poștală. Am selectat mesajul și am tastat Enter. Figura 19-1 arată ecranul monitorului în momentul în care elm a afișat mesajul poștal.

Figura 19-1

Citirea poștei cu programul elm.



Așa cum arată mesajul emis în urma invocării comenzii elm, răspunsurile de pe alte sisteme au ajuns și ele pe sistemul meu Linux.

Astfel, fișierul inițial de configurare pentru sendmail de pe CD-ROM-ul anexat acestei cărți trebuie să fie adecvat pentru trimiterea și primirea de e-mail, cu condiția ca sistemul dumneavoastră Linux să aibă o conexiune la Internet.

Mecanismul de livrare a poștei

Pe un calculator gazdă din Internet, agentul de transport pentru poștă sendmail livrează poșta folosind protocolul Simple Mail Transfer Protocol (SMTP). Documentație pentru SMTP se găsește în RFC 821, „Simple Mail Transfer Protocol,” Jonathan Postel, 1982.

Agenții de transport ai poștei care se bazează pe SMTP ascultă de portul TCP 25 și folosesc un mic set de comenzi text pentru a interacționa cu alți agenți ce transportă poșta. De fapt, comenzile sunt destul de simple, astfel încât le puteți folosi direct pentru a trimite un mesaj poștal.

Următorul exemplu arată cum folosesc eu comenzi SMTP pentru a trimite un mesaj poștal în contul meu de pe PC-ul Linux dintr-o sesiune telnet ce rulează pe alt sistem din rețeaua locală:

```
telnet lnbp200 25
Trying 192.168.1.200...
Connected to lnbp200
Escape character is '^]'.
220 lnbp200 ESMTP Sendmail 8.8.7/8.8.7; 16 May 1998
17:39:31-0400
HELO lnbp200
250 lnbp200 Hello lnbp75 [192.168.1.1], pleased to meet you
MAIL FROM: naba
250 naba... Sender ok
RCPT TO: naba
250 naba... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
```

Testing... 1 2 3

Sending mail by telnet to port 25

```
250 RAA00476 Message accepted for delivery
QUIT
221 lnbp200 closing connection
Conexiunea încheiata de un calculator gazda necunoscut.
```

Comanda telnet deschide o sesiune telnet pe portul 25 – portul unde sendmail așteaptă comenzi SMTP. Procesul sendmail de pe sistemul Linux răspunde imediat printr-un anunț.

Eu titez HELO lnbp200 pentru a inițializa o sesiune cu calculatorul gazdă numit lnbp200. Procesul sendmail răspunde cu un salut. Pentru a trimite mesajul poștal, încep prin comanda MAIL FROM:, care specifică expeditorul mesajului (Introduc numele utilizatorului pe sistemul de pe care trimit mesajul).

Apoi, folosesc comanda RCPT TO: pentru a indica destinatarul mesajului. Dacă doresc să trimit mesajul la mai mulți destinatari, trebuie doar să indic adresa fiecăruia în cadrul comenzii RCPT TO:.

Pentru a introduce mesajul poștal, folosesc comanda DATA. În replică la comanda DATA, sendmail afișează o instrucțiune prin care îmi indică faptul că trebuie să închei mesajul cu un punct pe o linie separată. Eu introduc mesajul și-l închei cu un singur punct pe o linie separată. Procesul sendmail afișează un text ce indică faptul că mesajul a fost acceptat pentru livrare. În cele din urmă, închei sesiunea cu sendmail prin comanda QUIT.

Apoi, intru în sistemul meu Linux și verific poșta prin comanda mail. Iată ce văd când afișez mesajul poștal pe care l-am trimis cu sendmail în cadrul sesiunii SMTP demonstrative:

```
mail
Mail version 5.5-kw 5/30/95. Type ? for help.
"/var/spool/mail/naba": 1 message 1 new
>N 1 nab@lnbp200 Sat May 16 17:40 12/376
& 1
Message 1:
From nab@lnbp200 Sat May 16 17:40:29 1998
Return-Path: <naba>
Received: from lnbp200 (lnbp75 [192.168.1.1])
    by lnbp200 (8.8.7/8.8.7) with SMTP id RAA00476
    for naba; Sat, 16 May 1998 17:39:50 -0400
Data: Sat, 16 May 1998 17:39:50 -0400
From: Linux Red Hat User <nab@lnbp200>
Message-Id: <199805162139.RAA00476@lnbp200>
Status: R
```

Testing... 1 2 3

Sending mail by telnet to port 25

După cum puteți vedea, comenzile SMTP sunt destul de simplu de înțeles. Acest exemplu ar trebui să vă ajute să înțelegeți cum folosește un agent de transfer al poștei protocolul SMTP pentru a transfera poșta pe Internet.

Grupuri de discuții

Grupurile de discuții își au originile în Usenet – o rețea de mesaje de tipul stochează-și-trimite (store-and-forward) pentru schimbul de poștă electronică și mesaje. Usenet lucrează ca un telegraf în care mesajele și poșta sunt trimise de la un sistem la altul. În Usenet sistemele nu sunt conectate în rețea; ele pur și simplu stabilesc conexiunea

dial-up și folosesc protocolul de copiere UNIX-to-UNIX (UNIX-la-UNIX Copy protocol – UUCP) pentru a transfera mesajele de text.



Observație

Usenet este o colecție de calculatoare conectate lejer, care a funcționat bine și continuă să fie folosită datorită cheltuielilor foarte mici necesare conectării la aceasta. Aveți nevoie doar de un modem și de un loc dispus să stocheze și să trimită poșta și mesajele dumneavoastră. Trebuie să setați UUCP pe sistemul dumneavoastră, dar nu aveți nevoie de o conexiune de rețea de durată; câteva legături telefonice sunt tot ce aveți nevoie pentru a continua fluxul de e-mail și mesaje. Dezavantajul este că nu puteți folosi serviciile TCP/IP, ca de exemplu World Wide Web, telnet sau ftp.

De la Usenet, grupurile de discuții au migrat către Internet. În locul UUCP, mesajele sunt acum transportate prin intermediul unui protocol Network News Transfer Protocol (NNTP), care este descris în RFC 977, „Network News Transfer Protocol: A Proposed Standard for Stream-Based Transmission of News”, B. Kantor și P. Lapsley, 1986.

Chiar dacă protocolul de transport al mesajelor s-a schimbat de la UUCP la NNTP, conceptul „stochează-și-trimite” (store-and-forward) al transferului de mesaje încă mai există. Astfel, dacă doriți să primiți mesaje pe sistemul dumneavoastră Linux, trebuie să găsiți un server de mesaje de unde sistemul dumneavoastră să le poată descărca.



Sugestie

Dacă aveți un contract cu un furnizor de servicii Internet (Internet Service Provider – ISP), atunci acesta ar trebui să vă ofere acces la un server de mesaje. Aceste servere de mesaje din Internet comunică folosind NNTP. Apoi, puteți folosi un cititor de mesaje cu facilități NNTP, ca de exemplu tin, pentru a accesa serverul de mesaje și a citi grupurile de discuții selectate. Acesta este modul cel mai simplu de a accesa mesaje pe PC-ul dumneavoastră Linux care este calculator gazdă în Internet.

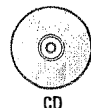
În discuția care urmează, voi presupune că ați obținut acces la un server de mesaje de la ISP-ul dumneavoastră.

Citirea mesajelor

Pentru a citi mesaje, aveți nevoie de un cititor de mesaje (*news reader*) – un program ce vă permite să selectați un grup de discuții și să examinați articolele existente acolo. Trebuie să înțelegeți, de asemenea, ierarhia grupului de discuții și convenția denumirilor, prezentate în secțiunea „Ierarhia grupurilor de discuții”. Mai întâi, vreau să vă arăt cum să citiți mesajele de pe un server de mesaje.

Ca și programele pentru poșta, sunt disponibile suficiente cititoare de mesaje. Cele mai multe cititoare de mesaje au facilitățile de *threading*, care înseamnă organizarea răspunsurilor la fiecare articol ca o secvență separată de articole. Dintre aceste programe, următoarele două sunt cele mai populare:

- Programul *threaded read news* (prezentator de mesaje după firul discuției), trn, care este varianta îmbunătățită a lui rn, venerabilul program cititor de mesaje. Acest cititor de mesaje este numit astfel pentru că interconectează un articol cu orice răspuns.
- Programul full-screen news reader (cititor de mesaje orientat spre pagină), tin. Ca și trn, tin are facilități de *threading*.



CD

CD-ROM-ul anexat acestei cărți conține câteva cititoare de mesaje, inclusiv trn și tin. Ambele cititoare de mesaje ar trebui să fie instalate, cu condiția să fi selectat componentul *Mail/WWW/News Tools* în timpul instalării Linux Red Hat.

Setarea variabilei de mediu NNTPSERVER

Pentru a citi mesaje de pe un server de mesaje folosind NNTP, trebuie să definiți variabila de mediu NNTPSERVER. Atribuiți acestei variabile numele serverului dumneavoastră de mesaje. Ambele cititoare de mesaje, tin și trn, examinează variabila de mediu NNTPSERVER pentru a afla numele serverului de mesaje pe care să-l contacteze.

De exemplu, numele serverului meu de mesaje este news.digex.net. Prin urmare, definesc variabila NNTPSERVER astfel:

```
NNTPSERVER=newsreader.digex.net; export NNTPSERVER
```

Crearea unui fișier .newsrc

Pe Internet există foarte multe grupuri de discuții. Când un cititor de mesaje își începe sesiunea de lucru, el trebuie să știe ce grupuri de discuții vreți să citiți. Ambele cititoare de mesaje, tin și trn, se așteaptă să găsească o listă de grupuri de discuții într-un fișier numit .newsrc din directorul dumneavoastră inițial.

Pentru a pregăti fișierul .newsrc, trebuie să știți cum să denumiți grupurile de discuții. Veți învăța mai multe despre denumirea grupurilor de discuții în secțiunea „Ierarhia grupurilor de discuții”.

Dacă vă interesează informații despre Linux, puteți crea un fișier .newsrc, specificând următoarele grupuri de discuții:

```
comp.os.linux.networking:
comp.os.linux.answers:
comp.os.linux.misc:
comp.os.linux.announce:
comp.os.linux.setup:
Comp.os.linux.hardware:
```

Fiecare linie arată numele unui grup de discuții, urmat de două puncte (:). După ce începeți să citiți, cititorul de mesaje completează numerele articolelor după numele grupului de discuții.

Rularea programului tin cu opțiunea -r pentru citirea mesajelor

Atât timp cât variabila de mediu NNTPSERVER este identică cu numele serverului dumneavoastră de mesaje, puteți folosi oricare dintre cititoare de mesaje – trn sau tin – pentru citire. Puteți încerca să citiți cu fiecare dintre cititoare de mesaje pentru a vă decide care vă place cel mai mult.

Pentru a citi mesaje cu tin folosiți următoarea comandă:

```
tin -r
tin 1.2 PL2 [UNIX] (c) Copyright 1991-93 Iain Lea.
Connecting to newsreader.digex.net...
Reading news active file...
Reading attributes file...
Reading newsgroup file...
```



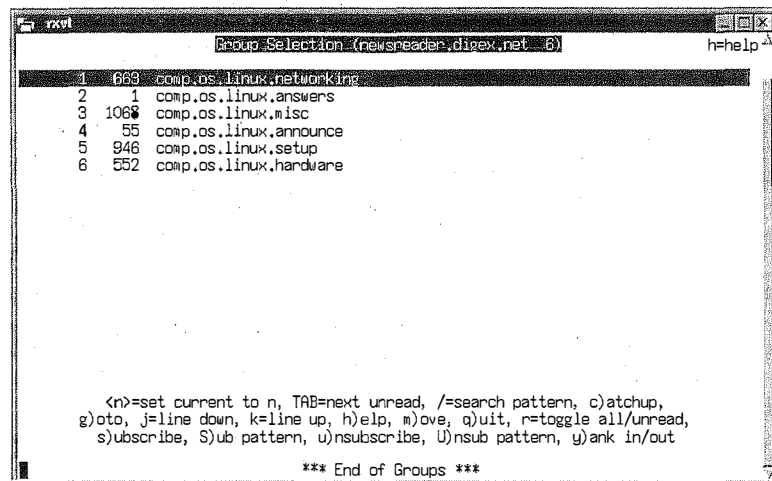
Sugestie

Opțiunea -r determină tin să citească mesaje de pe un server de mesaje de la distanță (cel pe care-l specificați prin variabila de mediu NNTPSERVER). După ce pornește, tin se conectează la serverul de mesaje și comunică folosind NNTP. Cititorul de mesaje tin afișează mesaje de stare pe măsură ce avansează către punctul în care adună grupurile de discuții listate în fișierul .newsrc din directorul dumneavoastră inițial.

După obținerea informațiilor despre grupurile de discuții de la serverul de mesaje, tin afișează acele grupuri de discuții într-un meniu text pe tot ecranul, ca în figura 19-2.

Figura 19-2

Un ecran tipic de selectare a grupurilor de discuții afișat de programul tin.



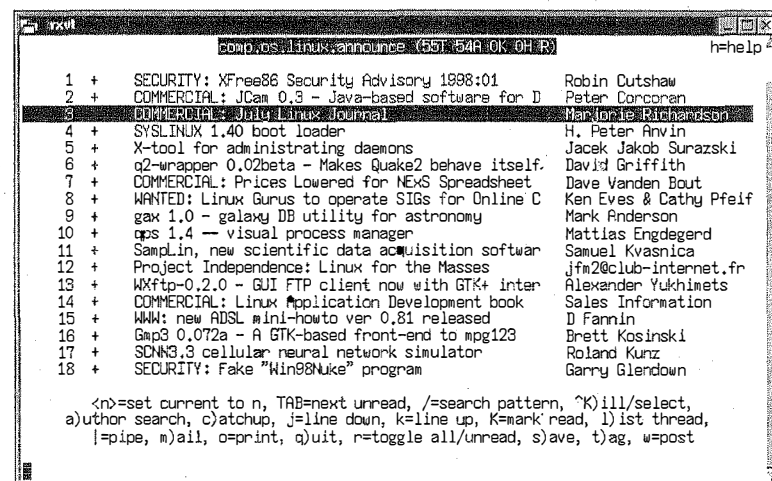
Lista de grupuri de discuții pe care o vedeți depinde, bineînțeles, de conținutul fișierului .newsrc din directorul dumneavoastră inițial. Figura 19-2 arată ecranul inițial tin pentru fișierul meu .newsrc.

În ecranul de selecție a grupurilor de discuții din programul tin, vedeți lista acestora în partea de sus a ecranului și informații ajutătoare în partea de jos. Inițial, este selectat primul grup de discuții (apare evidențiat). Tastați j pentru a vă deplasa în partea de jos a listei; tastați k pentru a vă deplasa în sus. (Într-adevăr, tastele pentru deplasare sunt aceleași ca la editorul v.) Puteți, de asemenea, apăsa tastele cu săgeți direcționale sus și jos pentru a selecta un nou grup de discuții.

După ce selectați un grup de discuții pentru citit, apăsați Enter pentru a vedea lista articolelor conținută de acesta. Figura 19-3 arată ecranul după ce am selectat grupul de discuții comp.os.linux.announce și am apăsat j pentru a derula în jos lista la un articol care mă interesa.

Figura 19-3

Lista articolelor dintr-un grup de discuții (comp.os.linux.announce) afișată de programul tin.



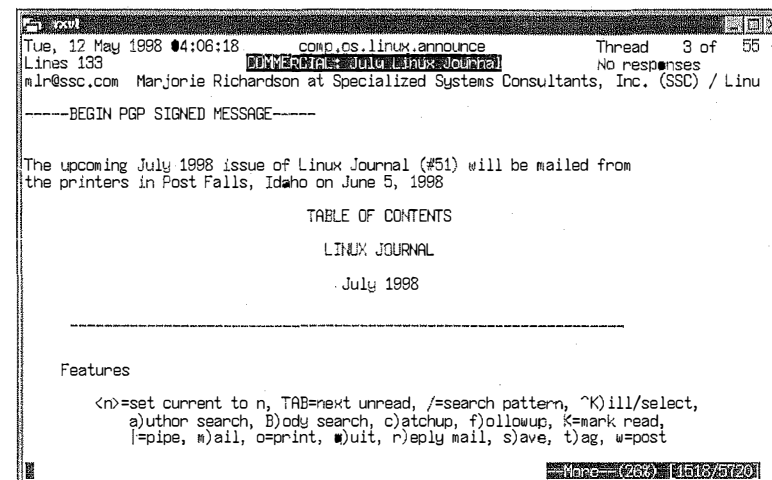
Așa cum arată și articolul evidențiat din figura 19-3, mă interesează un articol despre ediția din 1998 a publicației Linux Journal. Pentru a citi acest articol, tastez Enter. Cititorul de mesaje tin descarcă textul aferent articolului din serverul de mesaje și-l afișează, ca în figura 19-4.

Puteți citi textul aferent unui articol ecran cu ecran. Pentru a vedea ecranul următor, apăsați tasta spațiu sau tasta Page Down. Pentru a reveni la meniul de selecție a articolelor, tastați q.

Tastând încă o dată q ajungeți la ecranul de selecție a grupurilor de discuții. Pentru a părăsi programul tin, tastați q în ecranul de selecție a grupurilor de discuții.

Figura 19-4

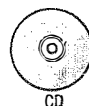
Cititorul de mesaje tin, afișând textul unui articol.



Observație

Cu toate că folosirea programului tin cu opțiunea -r pentru citirea mesajelor dintr-un server de mesaje de la distanță este „fără dureri“, poate dura un timp „dureros“ de lung până când cititorul de mesaje va descărca articolele mesajelor printr-o legătură lentă. Chiar dacă folosiți o conexiune PPP și un modem cu viteză de 56 Kbps, citirea mesajelor prin legături PPP este suficient de lentă încât să vă determine să doriți cel puțin o legătură T1 de 1,54 Mbps cu ISP-ul dumneavoastră

Folosirea unui alt cititor de mesaje: trn



CD

Mesajele se pot citi ușor și cu cititorul de mesaje trn. Pentru a citi mesaje dintr-un server de mesaje NNTP, vă trebuie o versiune de trn configurată pentru accesul mesajelor NNTP. Pachetul trn-nttp din setul N al CD-ROM-ului anexat acestei cărți conține versiunea corespunzătoare a trn.

După ce atribuiți variabilei de mediu NNTPSERVER numele serverului dumneavoastră de mesaje, rulați trn astfel:

trn

*** NEWS NEWS ***

Welcome to trn 3.6, which continues on the feature trail, adding support for slow net connections, and squashing some bugs. Trn is "threaded read news", based on rn.

You can type 'h' at any prompt to display a summary of the commands that are available. This message will not be displayed again unless it is updated with new information.

Wayne Davison <dasvison@borland.com.>

```
[Type space to continue]
Unread news in comp.os.linux.networking    665 articles
Unread news in comp.os.linux.answers       1 article
Unread news in comp.os.linux.misc          1072 articles
Unread news in comp.os.linux.announce      55 articles
Unread news in comp.os.linux.setup         949 articles
etc.
```

```
= = = = = 665 unread articles in comp.os.linux.networking - - read now?
```

```
[+ynq]
```

Getting overview file.

Cititorul de mesaje trn descarcă informația despre grupurile de discuții listate în fișierul .newsrc din directorul dumneavoastră inițial. Apoi afișează numărul de articole pentru câteva din grupurile de discuții.



Sugestie Datorită naturii aleatoare a Internetului, sunt adăugate continuu grupuri de discuții noi (în special cele al căror nume încep cu alt). Dacă fișierul dumneavoastră .newsrc nu conține numele unor grupuri de discuții adăugate recent, trn întreabă

dacă doriți să vă înscrieți în grupul de discuții. În general, eu tasez N (cu majuscule) pentru a indica faptul că nu doresc să adaug grupuri de discuții noi. După afișarea câtorva mesaje despre grupuri de discuții noi, trn vă solicită numele primului grup de discuții. Pentru a citi acest grup de discuții, tastați y sau Enter; pentru a-l omite, tastați n. Pentru a părăsi trn, tastați q.

Dacă selectați un grup de discuții de citit, trn vă oferă un fișier cu date generale pentru acel grup (este fișierul cu lista de articole ordonate după firul discuției, astfel încât fiecare articol este grupat cu răspunsurile lui). Apoi trn afișează lista articolelor din grupul de discuții selectat. Fiecare fir este identificat de o literă ce apare în prima coloană. Pentru a vedea un fir, folosiți tastele cu săgeți direcționale pentru a-i selecta litera și apoi tastați Enter. Cititorul de mesaje trn descarcă și afișează primul articol al celui fir.

În timp ce citiți articole, vă puteți deplasa printre ele folosind câteva comenzi trn. În general, comenzile sunt formate dintr-o singură literă.

Pentru a vedea comenzile trn, disponibile permanent, tastați h.

Dacă tastați q în timp ce citiți un articol, trn revine la nivelul de selectare a grupului de discuții și afișează un prompt, după cum urmează:

```
= = = = = 55 unread articles in comp.os.linux.announce - - read now?
```

```
[+Ynq]
```

Dacă tastați h în acest punct, trn afișează o listă de comenzi pentru selectarea articolelor.

Pentru a părăsi trn, tastați q la prompt-ul de selectare a articolului.

Ierarhia grupurilor de discuții

Articolele de mesaje sunt organizate într-o ierarhie de grupuri de discuții, pentru ușurința întreținerii și a folosirii. Un nume de grup de discuții tipic arată astfel:

Comp.os.linux.announce

Acest nume ne indică faptul că comp.os.linux.announce este un grup de discuții pentru anunțuri (announce) despre sistemul de operare Linux (os.linux) și ca aceste subiecte fac parte din vastul domeniu al calculatoarelor (comp).

După cum puteți vedea, formatul numelui unui grup de discuții este o succesiune de cuvinte, separate prin puncte. Aceste cuvinte desemnează ierarhia grupului de discuții.

Pentru a înțelege ierarhia grupurilor de discuții, comparați numele unui grup de discuții cu numele complet al unui fișier (ca de exemplu /usr/lib/news/bin/nntpget) din Linux. La fel cum calea fișierelor indică ierarhia directoarelor, numele grupurilor de discuții indică ierarhia acestora. În numele fișierelor, un slash (/) separă numele directoarelor; în numele unui grup de discuții, un punct separă diferitele niveluri ale ierarhiei grupului de discuții.

În numele unui grup de discuții, primul cuvânt reprezintă categoria grupului de discuții. Grupul de discuții comp.os.linux.announce, de exemplu, face parte din categoria comp, în timp ce alt.books.technical face parte din categoria alt.

Tabelul 19-1 prezintă câteva categorii importante de grupuri de discuții.

Tabelul 19-1 Categorii importante de grupuri de discuții

Categorie	Subiect
alt	Grupuri de discuții „alternative” (nesupuse nici unei reguli), cu subiecte de tot felul.
bionet	Grupuri de discuții despre biologie
bit	Grupuri de discuții despre rețeaua bitnet
biz	Grupuri de discuții despre afaceri
clari	Servicii pentru mesaje clarinet
comp	Grupuri de discuții despre software și hardware
ieee	Grupuri de discuții pentru Institutul de inginerie electrotehnică și electronică
k12	Grupuri de discuții dedicate școlii primare și secundare
misc	Grupuri de discuții diverse
news	Grupuri de discuții despre administrarea mesajelor Internet
rec	Grupuri de discuții recreative și despre artă
sci	Grupuri de discuții despre știință și inginerie
soc	Grupuri de discuții despre evenimente sociale și culturale
talk	Discuții despre probleme curente

Această scurtă listă de categorii este înșelătoare, pentru că nu conține marea varietate de grupuri de discuții disponibile pentru fiecare categorie. Deoarece fiecare categorie de grup de discuții conține câteva niveluri de subcategorii, numărul total al grupurilor de discuții ajunge la câteva mii. Numai categoria comp are mai mult de 500 de grupuri de discuții.

De obicei, trebuie să vă limitați la alegerea grupurilor de discuții conform intereselor dumneavoastră. Dacă sunteți interesați de Linux, de exemplu, puteți alege unul sau mai multe din următoarele grupuri de discuții:

■ comp.os.linux.admin: Informații despre administrarea sistemului Linux

■ comp.os.linux.advocacy: Discuții despre promovarea Linux

■ comp.os.linux.announce: Anunțuri importante despre Linux. Acest grup de discuții este moderat, adică trebuie să trimiteți articolul unui moderator, care îl afișează apoi în cadrul grupului de discuții

- `comp.os.linux.answers`: Întrebări și răspunsuri despre Linux. Toate documentele HOWTO din Linux (vezi capitolul 1 pentru o listă a documentelor HOWTO din Linux) sunt afișate în acest grup de discuții moderat
- `comp.os.linux.development`: Activitatea curentă pentru dezvoltarea Linux
- `comp.os.linux.development.apps`: Dezvoltarea aplicațiilor Linux
- `comp.os.linux.development.system`: Dezvoltarea sistemului de operare Linux
- `comp.os.linux.hardware`: Discuții despre Linux și diferite tipuri de hardware
- `comp.os.linux.help`: Asistență pentru diferite aspecte din Linux
- `comp.os.linux.misc`: Diverse subiecte legate de Linux
- `comp.os.linux.networking`: Lucrul în rețea sub Linux
- `comp.os.linux.setup`: Instalarea și setarea Linux
- `comp.os.linux.x`: Discuții despre setarea și rularea X Window System sub Linux

Trebuie să fiți selectiv în privința grupurilor de discuții pe care le citiți, deoarece este imposibil să fiți la curent cu toate noutățile, chiar și într-o zonă bine definită cum este Linux. Când instalați și setați pentru prima dată Linux, puteți citi grupuri de discuții ca de exemplu `comp.os.linux.setup`, `comp.os.linux.hardware` și `comp.os.linux.x` (în special dacă rulați X). După ce ați instalat și rulat Linux, puteți învăța numai lucrurile noi, specifice sistemului Linux. Pentru astfel de informații ar trebui să citiți grupul de discuții `comp.os.linux.announce`.

Înscrierea în grupurile de discuții

Spre deosebire de reviste și ziare, în grupurile de discuții nu este necesar să vă înscrieți efectiv; puteți citi orice grup de discuții disponibil pe serverul de mesaje. Administratorul serverului de mesaje poate decide să excludă anumite grupuri de discuții, pe care bineînțeles nu le mai puteți citi.

Singurul lucru similar cu înscrierea este fișierul `.newsrc` din directorul dumneavoastră inițial. Toate cititoarele de mesaje consultă acest fișier pentru a determina ce grupuri de discuții doriți să citiți. Din interiorul cititorului de mesaje puteți folosi o comandă precum `g`, urmată de numele grupului de discuții în care doriți să vă înscrieți (în cititorul de mesaje `trn`). Când vă înscrieți într-un grup de discuții, cititorul de mesaje adaugă, pur și simplu, numele acestuia în fișierul `.newsrc`.

Afișarea mesajelor (posting)

Puteți folosi un cititor de mesaje pentru a afișa un articol de mesaj (un articol nou sau un răspuns la un afiș mai vechi) într-unul sau mai multe grupuri de discuții. Comanda exactă pentru afișarea unui articol depinde de cititorul de mesaje. În cititorul de mesaje `trn`, parcurgeți următorii pași pentru a afișa un articol:

1. În timp ce citiți un articol în `trn`, tasteați `f`. Cititorul de mesaje vă întreabă dacă afișați un subiect care nu are legătură cu articolul pe care-l citeați în momentul când ați apăsă `f`. Tasteați `y` pentru Yes, astfel:

Are you starting an unrelated topic? [ynq] **y**

2. Apoi, cititorul de mesaje vă solicită să introduceți subiectul noului afiș și distribuția. Pentru această afișare de probă, tasteați cuvântul *ignore* în rândul ce conține subiectul. Altfel, oricine primește articolul dumneavoastră vă va răspunde prin poștă că acesta a ajuns la el; acesta este

scopul grupului de discuții `misc.test`. Pentru distribuție, tasteați `na` pentru a indica North America (America de Nord), după cum urmează:

Subject: ignore no reply test
Distribution: **na**

3. Cititorul de mesaje folosește programul `Pnews` pentru a afișa un articol în grupuri de discuții. Dacă afișați un articol pentru prima dată, vedeți următorul mesaj:

(leaving cbreak mode; cwd=/usr/etc)

Observați că nu ați mai folosit niciodată această versiune a programului `Pnews`. Va voi acorda asistență suplimentară acum, când este prima dată, dar apoi trebuie să știți ce ați învățat. Dacă nu înțelegeți vreo întrebare, tasteați `h` și un `CR` (carriage return – retur de car) pentru ajutor. Dacă nu ați mai afișat până acum un articol pe rețea, este recomandat să citiți documentul ce se găsește în `news.announce.newusers` astfel încât să puteți evita cele mai uzuale greseli. Pentru a face acest lucru, întrerupeți `Pnews`, poziționați-vă pe promptul corespunzător nivelului cel mai de sus al programului `trn` și folosiți comanda `"g news.announce.newusers"` pentru a vă deplasa la acel grup.

4. După mesaj, cititorul de mesaje vă întreabă dacă doriți într-adevăr să afișați un articol. Tasteați `y` pentru a continua.

Acest program poate afișa mesaje pe multe tipuri de calculatoare. Sunteți absolut sigur că asta doriți? [ny] **y**

5. Cititorul de mesaje întreabă dacă aveți un fișier pe care doriți să-l includeți ca articol de mesaj, utilizând următorul prompt:

Prepared file to include [none]:

Tasteați `Enter` pentru a continua.

6. Cititorul de mesaje vă solicită să introduceți numele editorului dumneavoastră (cel prestabilit este `vi`) cu următoarea explicație:

Un fișier temporar a fost creat pentru dumneavoastră pentru a-l edita. Asigurați-vă că ați lasat cel puțin o linie goală între antetul și corpul mesajului dumneavoastră (și până când o anumită defecțiune este reparată la nivelul întregii rețele, nu începeți corpul mesajului dumneavoastră cu nici o indentare, pentru că ar putea fi anulată).

În cadrul antetului pot exista câmpuri pe care nu le înțelegeți. Dacă nu înțelegeți un câmp (sau chiar dacă-l înțelegeți), puteți pur și simplu să-l lasați gol, deoarece va dispărea atunci când articolul este afișat.

Tasteați `Return` pentru a obține editorul prestabilit sau tasteați numele editorului dumneavoastră preferat.

Editor [/usr/bin/vi]:

Tasteați `Enter` pentru a compune articolul de mesaj cu editorul `vi`.

7. Cititorul de mesaje pornește editorul `vi`. În editor, puteți introduce numele grupului de discuții și orice alte câmpuri în articolul de mesaj (inclusiv corpul articolului). Iată ce am completat eu (cu caractere albine):

Newsgroup: **misc.test**
Subject: ignore no reply test
Summary:
Expires:

Sender:
Followup-To:
Distribution: na
Organization:
Keywords: **test ignore**
Cc:

Testing ignore

Observați că nu este nici o intrare pentru numele organizației. Puteți edita acest câmp pentru a indica numele organizației dumneavoastră sau definiți variabila de mediu ORGANIZATION (atribuiți-i numele organizației dumneavoastră) înainte de a porni trn. După ce terminați editarea articolului, salvați folosind comanda :wq.

8. Cititorul de mesaje vă prezintă numele (și o descriere) grupului de discuții în care urmează să afișați articolul. Apoi trn vă solicită o comandă. Tastați s pentru a trimite articolul. Interacțiunea cu cititorul de mesaje decurge astfel:

```
Your article's newsgroup (grupul de discutii al articolului dumneavoastra):
Misc.test      [no description available / nici o descriere disponibila]
```

Verificarea ortografiei, Trimite, Renunta, Editeaza sau Listeaza?

9. Cititorul de mesaje trn returnează articolul pe care-l citeați în momentul când ați început să afișați acest articol. Tastați q pentru a părăsi grupul de discuții pe care-l citeați.

10. Tastați g misc.test pentru a vă înscrie în grupul de discuții misc.test (unde tocmai ați afișat noul articol).

11. Examinați la cel mai recent articol din fișierul misc.test, care ar trebui să fie articolul afișat de dumneavoastră

A plecat articolul mai departe?

Dacă afișați un articol și citiți grupul de discuții imediat, veți vedea noul articol, dar aceasta nu înseamnă că articolul a ajuns la alte situri din Internet. La urma urmei, articolul afișat de dumneavoastră apare pe serverul dumneavoastră de mesaje imediat, deoarece chiar acolo ați afișat articolul. Datorită modelului store-and-forward de distribuție a mesajelor, articolul de mesaj se propagă treptat de la serverul dumneavoastră de mesaje către alte servere din întreaga lume.



Secret

Grupul de discuții misc.test furnizează o modalitate de a verifica dacă mesajele afișate de dumneavoastră se propagă pe Internet. Dacă afișați un articol în acest grup de discuții și nu includeți cuvântul *ignore* în cadrul subiectului, serverele de mesaje vor confirma primirea articolului prin trimiterea unui e-mail la adresa indicată în câmpul Reply-To din antetul articolului.

Dacă sistemul dumneavoastră Linux este conectat la Internet, încercați să afișați un articol în grupul de discuții pentru a verifica dacă acesta se propagă în rețea. Ar trebui să fiți pregătit să primiți zeci de răspunsuri din diferite situri, care să confirme primirea articolului dumneavoastră.

FTP anonim sigur

Pe lângă poșta și mesaje, FTP anonim este un serviciu uzual pe un calculator gazdă din Internet. Poate sunteți familiarizat cu FTP (File Transfer Protocol), pe care-l puteți folosi pentru transferul fișierelor de pe un sistem pe altul. Când folosiți FTP pentru transferul fișierelor către sau de la un calculator de la distanță, trebuie să vă conectați la sistemul de la distanță înainte de a folosi FTP.



Referință încrucișată

Serviciul FTP anonim se referă la folosirea numelui de utilizator *anonymous*, pe care-l poate folosi oricine pentru a transfera fișiere dintr-un sistem prin intermediul FTP. FTP anonim este o modalitate obișnuită de partajare a fișierelor în Internet. Capitolul 2, „Actualizarea sistemului Linux”, prezintă modul de utilizare a FTP anonim pentru a descărca părți din nucleele Linux din calculatorul gazdă din Internet denumit ftp.funet.fi.

Dacă ați folosit FTP anonim pentru a descărca fișiere de pe diferite situri din Internet, cunoașteți deja facilitățile acestui serviciu. Prin intermediul său, puteți pune la dispoziția oricui anumite informații din Internet. Dacă aveți o nouă aplicație Linux pe care vreți s-o partajați, setați FTP anonim pe PC-ul dumneavoastră Linux și plasați programul într-un director potrivit. După aceea, trebuie doar să anunțați pe toata lumea (probabil printr-un afiș în grupul de discuții comp.os.linux.announce) că aveți un nou program disponibil. Acum oricine poate obține programul de pe sistemul dumneavoastră oricând dorește.

Chiar dacă conduceți o întreprindere pentru profit, puteți folosi serviciul FTP anonim în sprijinul clienților dumneavoastră. Dacă vindeți produse software sau hardware, probabil că doriți să oferiți informații tehnice sau soluții software prin intermediul acestui serviciu.

Din păcate, avantajele serviciului FTP anonim își au prețul lor. Dacă nu-l configurați corect, intruși și poștași pot accesa sistemul dumneavoastră. Unii intruși pot folosi discul dumneavoastră ca spațiu de depozitare temporară pentru diferite fișiere; alții vă pot umple discul cu fișiere vechi, scoțând astfel sistemul dumneavoastră din funcțiune. La cealaltă extremă, un intrus poate obține acces la nivel de utilizator (sau, mai rău, la nivel de rădăcină) în sistemul dumneavoastră și vă poate provoca mult mai multe pagube.



Observație

Dacă ați instalat Linux Red Hat de pe CD-ROM-ul anexat acestei cărți, aveți deja instalat FTP anonim pe sistemul dumneavoastră. Setarea prestabilită implică, de asemenea, precauțiile necesare în ceea ce privește siguranța.

În continuare sunt prezentate caracteristicile cele mai importante ale setării serviciului de FTP anonim:

■ Există un utilizator numit ftp ce se află în directorul /home/ftp. Utilizatorul nu are atribuit nici un program de tip shell. Iată ce obțineți când căutați ftp în fișierul /etc/passwd:

```
grep ftp /etc/passwd
ftp:*:14:50:FTP User:/home/ftp:
```

Observați asteriscul din câmpul al doilea, ceea ce înseamnă că nimeni nu se poate conecta cu numele de utilizator ftp.

■ Iată setarea pentru permisiune completă și informații despre proprietarul directorului /home/ftp:

```
dwxr-xr-x 6 root root 1024 Mar 11 14:58 ftp
```

Așa cum arată această linie, directorul /home/ftp este proprietatea utilizatorului root și permisiunea este setată pe valoarea 755 (numai rădăcina are acces la citire și scriere; ceilalți au acces numai la citire).

■ Puteți vedea conținutul directorului /home/ftp utilizând comanda ls -la. Rezultatul este următorul:

```
total 6
drwxr-xr-x 6 root root 1024 Mar 11 14:58 .
drwxr-xr-x 6 root root 1024 Feb 6 1996 ..
d--x--x--x 2 root root 1024 Mar 11 14:58 bin
```

```
d--x--x--x 2 root root 1024 Mar 11 14:58 etc
drwxr-xr-x 2 root root 1024 Mar 11 14:58 lib
dr-xr-sr-x 2 root root 1024 Nov 6 1997 pubs
```

Observați că setările pentru permisiunile directorilor bin și etc sunt 111 (numai execuție). Toate fișierele din interiorul directorului bin permit accesul numai pentru a fi executate (setare de permisiune 111). Toate fișierele din directorul etc permit accesul numai pentru citire (setare de permisiune 444).

■ Directorul pub este acela în care plasați orice fișier ce poate fi descărcat din sistemul dumneavoastră prin intermediul serviciului de FTP anonim.

Mai multe despre securitatea în Internet

Cele mai recente informații despre securitatea unui calculator gazdă din Internet sunt disponibile chiar pe Internet. Pentru această informație și pentru a fi la curent cu cele mai recente probleme de securitate, vizitați pagina organizației CERT (<http://www.cert.org/>) și citiți sfaturile, buletinele și rezumatele afișate periodic în situl FTP al organizației CERT (<ftp://info.cert.org/pub/>). (CERT era prescurtarea de la Computer Emergency Response Team – Echipa de intervenție pentru

calculatoare, dar în prezent este pur și simplu CERT – marcă de servicii a universității Carnegie Mellon).

Altă sursă bună de informații este Computer Incident Advisory Capability (CIAC) a Departamentului pentru Energie al SUA. Pe pagina lor de Web de la adresa <http://ciac.llnl.gov/>, veți găsi multe resurse despre securitatea în Internet.

Rezumat

Creșterea popularității rețelei Internet a avut ca rezultat răspândirea furnizorilor de servicii Internet (Internet Service Provider – ISP) pe tot cuprinsul Statelor Unite și în întreaga lume. Serviciile online, ca de exemplu America Online și CompuServe, oferă acum poșta Internet și navigare Web, aducând și mai mulți utilizatori pe Internet. Datorită facilităților pentru TCP/IP – limba universală al Internetului – un PC Linux este ideal ca sistem gazdă pentru Internet. Acest capitol descrie serviciile tipice disponibile pe un calculator gazdă din Internet și va arată cum să configurați aceste servicii – poșta, mesaje și FTP anonim pe un PC Linux.

După lectura acestui capitol, ați învățat următoarele:

- ▶ Cu toate că Internet este în primul rând „rețeaua rețelilor”, ea se manifestă diferit în funcție de modul de utilizare a facilităților acesteia. Pentru cei mai mulți oameni, Internetul este World Wide Web și poșta electronică (e-mail), în timp ce mulți alții o folosesc pentru a citi grupuri de discuții. Alții folosesc aplicațiile Internet ca de exemplu telnet și ftp pentru a accesa calculatoare de la distanță.
- ▶ World Wide Web, sistemul e-mail, grupurile de discuții și serviciul de FTP anonim sunt câteva dintre serviciile disponibile pe un calculator gazdă din Internet. Acest capitol descrie sistemul e-mail, grupurile de discuții și serviciile FTP anonim disponibile pe un PC Linux (World Wide Web este prezentat în capitolul următor).
- ▶ Programul pentru e-mail este format din două părți: un agent ce transportă poșta, care trimite și primește fizic mesajele poștale, și un agent ce utilizează poșta care citește mesajele și le pregătește pe cele noi.

- ▶ CD-ROM-ul anexat acestei cărți include câțiva agenți ce transferă poșta și agenți ce utilizează poșta. Acest capitol descrie cum să folosiți programul sendmail ca agent ce transportă poșta.
- ▶ Configurarea programului sendmail este complexă, dar puteți folosi fișierul dat ca exemplu pe CD-ROM. Tot ce trebuie să faceți este să setați corespunzător numele calculatorului dumneavoastră gazdă Linux.
- ▶ Pentru a citi mesajele de poșta electronică, puteți folosi programele mail sau elm.
- ▶ Grupurile de discuții își au originea în Usenet, care este o rețea de tipul store-and-forward. Mesajele călătoresc în jurul lumii dintr-un sistem în altul. În prezent, mesajele sunt transportate pe Internet prin intermediul protocolului Network News Transport Protocol (NNTP).
- ▶ Deoarece există mii de grupuri de discuții, stocarea numeroaselor articole de mesaje ocupă mult spațiu pe disc. De asemenea, articolele trebuie șterse periodic; altfel, discul se va umple. Cel mai bine este să citiți mesaje dintr-un server de mesaje administrat de un furnizor de servicii Internet.
- ▶ CD-ROM-ul anexat acestei cărți include câteva cititoare de mesaje, ca de exemplu tin și trn. Acest capitol vă arată cum să folosiți aceste cititoare pentru a citi mesaje dintr-un server de mesaje dedicat.
- ▶ Pentru a citi grupuri de discuții de pe un server de mesaje, ar trebui să atribuiți variabilei de mediu NNTPSERVER numele serverului de mesaje.
- ▶ FTP anonim este alt serviciu popular din Internet pentru distribuirea fișierelor. Folosind FTP anonim, oricine poate folosi serviciile FTP cu numele de utilizator anonymous și poate descărca fișiere din sistemul dumneavoastră. Cu toate că acest serviciu este folosit pentru distribuția datelor, el implică un risc de securitate dacă nu este setat corespunzător.
- ▶ Instalarea prestabilită a sistemului Linux include un server FTP cu facilități pentru serviciul FTP anonim. Setarea prestabilită pentru FTP anonim include precauțiile necesare de securitate.

Capitolul 20

Rularea unui server World Wide Web în Linux

În acest capitol

- ▶ Veți înțelege sistemul World Wide Web
- ▶ Veți înțelege adresele URL și legăturile de hipertext
- ▶ Veți înțelege protocolul HTTP
- ▶ Veți naviga în rețea
- ▶ Veți descărca și instala browserul de Web Netscape
- ▶ Veți configura serverul de Web Apache

Capitolele 18 și 19 v-au arătat cum să vă conectați PC-ul Linux la Internet (prin intermediul unui furnizor de servicii Internet) și cum să setați servicii uzuale din Internet – ca de exemplu poștă, mesaje și FTP anonim – pe sistemul dumneavoastră. Acest capitol revine la unul din motivele pentru care Internetul a devenit atât de popular în ultimii ani: World Wide Web (WWW, W3 sau pur și simplu Web), care furnizează o modalitate grafică simplă pentru a naviga și a accesa informații din Internet.

Fiind calculator gazdă în Internet, aveți nevoie doar de un browser de Web – o aplicație care „știe” cum să descarce și să afișeze documente Web – astfel că puteți începe să beneficiați de facilitățile Web. Puteți de asemenea să puneți informații la dispoziția altor utilizatori, prin pagini de Web – termenul comun pentru documente Web.

Acest capitol explică mai întâi ce este World Wide Web și descrie Hypertext Transfer Protocol (HTTP) – protocolul pentru schimbul de informații care determină rețeaua Web să funcționeze. Apoi vi se arată cum să descărcați browserul de Web Netscape, cum să-l setați și să-l utilizați.

Apoi, este descris modul de folosire a sistemului Web ca modalitate frecvent întâlnită de publicare a informațiilor pe Internet. Linux Red Hat conține serverul HTTP Apache (aceasta este denumirea tehnică pentru un server Web), pe care-l puteți folosi pur și simplu prin plasarea fișierelor dumneavoastră HTML în directorul corespunzător. Vă voi arăta unde ajung diferite fișiere și cum să configurați serverul.

Descoperirea World Wide Web

Dacă ați utilizat un server de rețea de orice fel, cunoașteți posibilitatea de a accesa fișiere ce se află la o locație partajabilă. Folosind o aplicație care procesează texte ce rulează pe calculatorul dumneavoastră, puteți deschide cu ușurință un document localizat pe server.

Acum imaginați-vă un procesor de text care vă permite să deschideți și să vedeți un document de pe orice calculator din Internet. Puteți vedea documentul în toată splendoarea

lui, cu text formatat și grafică. Dacă în document se face o trimitere la alt document (care poate fi amplasat pe un pe alt calculator), puteți deschide acel document legat executând clic pe referința lui. Acest tip de acces facil la documente distribuite este principala facilități pe care o oferă World Wide Web.

Bineînțeles, documentele trebuie să se găsească în formatul standard, astfel încât orice calculator (cu programul de Web adecvat) să poată accesa și interpreta documentul. În plus, un protocol standard este necesar pentru transferarea documentelor Web dintr-un sistem în altul.

Formatul standard al documentului Web este Hypertext Markup Language (HTML), iar protocolul standard pentru schimbul de documente Web este Hypertext Transfer Protocol (HTTP).



Observație

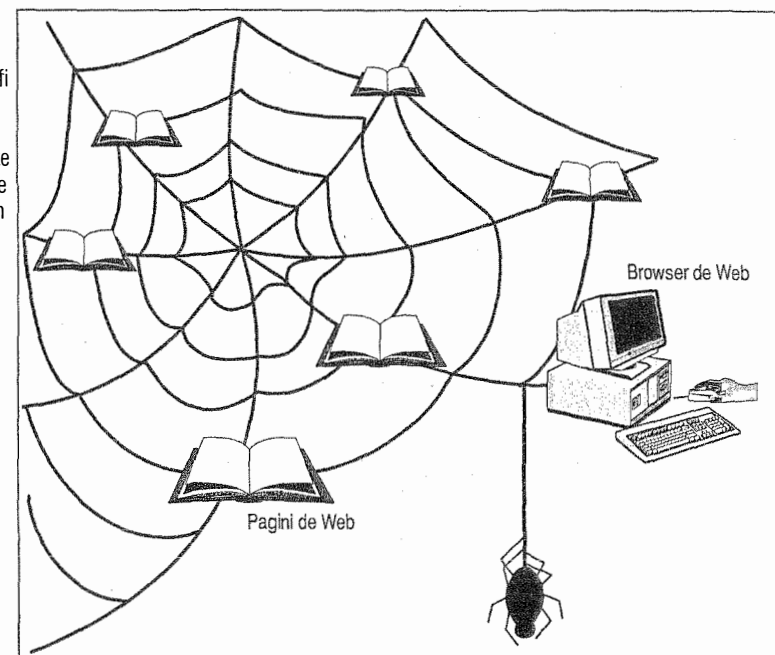
Un server Web este programul care furnizează documente HTML oricărui client care face cererea HTTP adecvată. Un browser de Web este programul client care descarcă un document HTML dintr-un server de Web și afișează conținutul grafic.

Ca pânza unui păianjen gigantic

World Wide Web este combinația dintre servere Web și documentele HTML ce conțin informația. Imaginați-vă rețeaua Web ca o carte gigantică, ale cărei pagini sunt împrăștiate pe Internet. Folosiți un browser de Web ce rulează pe calculatorul dumneavoastră pentru a vedea paginile, ca în figura 20-1.

Figura 20-1

Sistemul World Wide Web poate fi asemănat cu milioane de pagini, împrăștiate prin rețea, pe care le puteți folosi din calculatorul dumneavoastră folosind un browser de Web.



Sistemul World Wide Web poate fi asemănat cu milioane de pagini, împrăștiate prin rețea, pe care le puteți folosi din calculatorul dumneavoastră folosind un browser de Web.

Așa cum arată figura 20-1, paginile – documentele HTML – sunt legate prin conexiuni de rețea care seamănă cu o pânză de păianjen uriaș, astfel că puteți vedea de ce Web este denumit Web (web = pânză de păianjen). Cuvintele „World Wide” vin de la faptul că paginile de Web sunt legate în jurul lumii.

Legături și adrese URL

Ca și paginile cărților reale, paginile Web conțin text și grafice. Spre deosebire de cele din cărți, paginile Web pot conține informații multimedia, precum videoclipuri, sunete digitale și legături cu alte pagini de Web, prin intermediul cărora utilizatorul poate accesa imediat acele pagini.

Legăturile (*links*) dintr-o pagină de Web sunt referințe la alte pagini și le puteți folosi pentru a vă deplasa dintr-o pagină în alta. Browserul de Web afișează aceste legături ca text subliniat (cu o culoare diferită) sau ca imagini. Fiecare legătură este ca o instrucțiune pentru cititor – ceva de genul „pentru mai multe informații consultați capitolul 20”, pe care o puteți găsi într-o carte reală. Într-o pagină de Web nu trebuie decât să executați clic pe legătură și browserul de Web vă aduce pe ecran pagina corespunzătoare acestei legături, chiar dacă se află pe un calculator diferit.



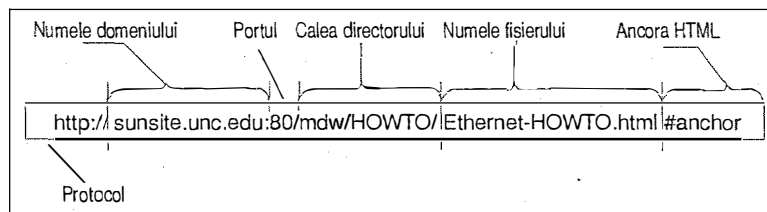
Observație

Termenul *hipertext* se referă la organizarea neliniară a textului (opusă organizării secvențiale, liniară a textului dintr-o carte sau revistă). Legăturile dintr-o pagină de Web sunt denumite legături de hipertext deoarece, executând clic pe o legătură, puteți „sări” la altă pagină, ceea ce este un exemplu de organizare neliniară a textului.

Această aranjare ridică o întrebare. Într-o carte reală, puteți cere cititorului să meargă la un anumit capitol sau pagină din carte. Dar cum indică o legătură de hipertext locația paginii de Web la care face trimitere? În World Wide Web, fiecare pagină are un nume special, numit adresă URL (*Uniform Resource Locator*). O adresă URL specifică în mod unic locația unui fișier dintr-un calculator, ca în figura 20-2.

Figura 20-2

Diferite părți componente ale adresei URL.



Așa cum ilustrează și figura 20-2, o adresă URL are următoarea secvență de componente:

1. Protocolul. Primul câmp este numele protocolului folosit pentru accesarea datelor localizate în fișierul specificat de URL. În figura 20-2, protocolul este `http://`, ceea ce înseamnă că adresa URL specifică locația unei pagini de Web. În continuare, sunt prezentate tipurile de protocole uzuale și semnificația lor:

- `file://` specifică numele unui fișier local ce va fi deschis și afișat. Puteți folosi acest fișier pentru a vedea fișiere HTML, fără să fie necesar să vă conectați la Internet.
- `ftp://` specifică un fișier accesibil prin intermediul protocolului FTP (File Transfer Protocol).
- `gopher://` specifică un document și numele unui calculator gazdă ce activează serverul Gopher de pe care va fi încărcat documentul. (Serverele Gopher furnizează informații folosind un protocol simplu.)

- `http://` specifică un document HTML accesibil prin intermediul protocolului HTTP (Hypertext Transfer Protocol).
- `mailto://` specifică o adresă de e-mail ce ar trebui folosită pentru a trimite un mesaj.
- `news://` specifică un grup de discuții pentru citire prin intermediul protocolului Network News Transfer Protocol (NNTP).
- `telnet://` specifică numele unui utilizator și al unui sistem pentru deschiderea unei sesiuni de lucru la distanță.
- `wais://` specifică numele unui sistem WAIS (Wide Area Information Server) de unde informația trebuie încărcată.

2. Numele domeniului. Această parte a adresei URL conține numele complet al domeniului calculatorului unde este localizat fișierul specificat de acest URL. Puteți, de asemenea, specifica o adresă IP în acest câmp (vezi capitolul 16 pentru mai multe informații despre adresele IP). Numele domeniului nu face distincția între majuscule și literele mici.

3. Adresa portului. Aceasta este adresa portului pentru serverul care implementează protocolul listat în prima parte a adresei URL (vezi capitolul 16 pentru detalii despre porturi de adrese). Această parte a adresei URL este opțională, deoarece există porturi prestabilite pentru toate protocolele. Portul prestabilit pentru HTTP, de exemplu, este 80. Totuși, unele situri pot configura serverul Web să asculte de alt port. În acest caz, adresa URL trebuie să conțină adresa portului.

4. Calea directorului. Acest câmp conține calea directorului pentru fișierul la care se face referire în URL. Pentru pagini de Web, acest câmp conține calea fișierului HTML. Calea directorului face distincția între majuscule și literele mici.

5. Numele fișierului. Acest câmp conține numele fișierului. Pentru pagini de Web, numele de fișier se termină de obicei cu `.htm` sau `.html`. Dacă omiteți numele fișierului, serverul Web returnează un fișier prestabilit (deseori denumit `index.html`). Numele fișierului face distincția între majuscule și literele mici.

6. Ancora HTML. Această parte opțională a adresei URL este utilizată pentru a determina browserul de Web să facă un salt la o anumită locație în fișier. Dacă această parte începe cu un semn de întrebare (?) în loc de semnul diez (#), textul ce urmează după semnul întrebării este considerat ca o întrebare. Serverul Web returnează informațiile bazate pe astfel de întrebări.

Protocolul de transfer prin hipertext (Hypertext Transfer Protocol)

Hypertext Transfer Protocol (HTTP) – protocolul care stă la baza sistemului World Wide Web – este denumit *Hypertext* pentru că paginile de Web conțin legături bazate pe hipertext. *Transfer Protocol* se referă la convențiile standard pentru transferul unei pagini Web prin rețea de la un calculator la altul. Cu toate că nu trebuie să înțelegeți protocolul HTTP pentru a seta un server Web sau pentru a folosi un browser Web, cred că veți considera instructiv să aflați cum lucrează rețeaua Web.

Înainte de a vă explica HTTP, ar trebui să luați în prealabil contact cu acesta. Pe cele mai multe sisteme, serverul Web ascultă de portul 80 și răspunde la orice solicitare HTTP trimisă aceluia port. Prin urmare, puteți folosi programul *telnet* pentru a vă conecta la portul 80 al unui sistem (care are un server Web) pentru a testa unele comenzi HTTP.

Pentru a vedea cum lucrează HTTP, parcurgeți următoarele etape:

1. Asigurați-vă că este stabilită conexiunea la Internet a PC-ului dumneavoastră Linux. (Dacă folosiți legături SLIP sau PPP, verificați dacă este stabilită conexiunea.)

2. Tastați următoarea comandă:

```
telnet www.idgbooks.com 80
```

3. După ce vedeți mesajul **Connected...**, tastați următoarea comandă HTTP:

```
GET / HTTP/1.0
```

și apăsați Enter de două ori. Ca răspuns la această comandă HTTP, serverul Web returnează câteva informații folositoare, urmate de conținutul fișierului prestabilit HTML (denumit de obicei index.html).

Iată ce am obținut când am testat comanda GET pe situl Web IDG Books:

```
telnet www.idgbooks.com 80
Trying 206.80.51.140...
Connected to leland.idgbooks.com.
Escape character is '^]'.
GET / HTTP/1.0
.....(Apasati Enter inca o data pentru a trimite o linie goala)
HTTP/1.1 200 OK
Date: Sun, 17 May 1998 22:35:19 GMT
Server: Apache/1.3b3
Connection: close
Content-Type: text/html
```

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
```

..... (linii sterse)

```
<title> IDG Books Online</title>
</head>
```

```
<body background="http: / / www.idgbooks.com/ /images /background.5.gif "
text="#0000
00" link="#0201FE" vlink="#000087" alink="# FF0000">
```

.....(linii sterse)

```
</body>
```

```
<html>
Conexiunea incheiata de un sistem gazda necunoscut.
```

Când încercați acest exemplu cu telnet, puteți vedea cu exactitate ce anume returnează serverul Web către browserul Web. Primele linii sunt informații administrative pentru browser. Serverul returnează următoarea informație:

■ O linie care arată că serverul folosește protocolul HTTP versiunea 1.1 și un cod de stare cu valoarea 200 indicând succes: HTTP/1.1 200 OK

■ Data și ora curentă. Un model de șir indicând ora și data arată astfel:

```
Date: Sun, 17 May 1998 22:35:19 GMT
```

■ Numele și versiunea programului de pe serverul Web. De exemplu, pentru un sit care rulează Apache HTTPD Version 1.3b3, serverul returnează următorul șir: Server: Apache/1.3b3

■ Tipul documentului returnat de serverul Web. Pentru documente HTML, tipul conținutului este raportat astfel:

```
Content-type: text/html
```

Documentul însuși urmează după informațiile administrative. Un document HTML are următorul aspect general:

```
<title> Aici se scrie titlul documentului</title>
<html>
<Aici se scriu attribute optionale ale corpului>
... Restul documentului se scrie aici
</body>
</html>
```

Puteți identifica această aranjare examinând lista care arată datele returnate de serverul Web ca răspuns la comanda GET. Deoarece exemplul folosește o comandă telnet pentru a obține documentul, conținutul HTML apare sub formă de linii de text. Dacă ați accesa aceeași adresă URL (<http://www.idgbooks.com>) cu un browser Web (ca de exemplu Netscape Navigator), ați vedea pagina în forma ei grafică, ca în figura 20-3.



Dacă nu aveți un browser de Web, învățați cum să descărcați și să setați o astfel de aplicație în următoarele secțiuni.

Observație

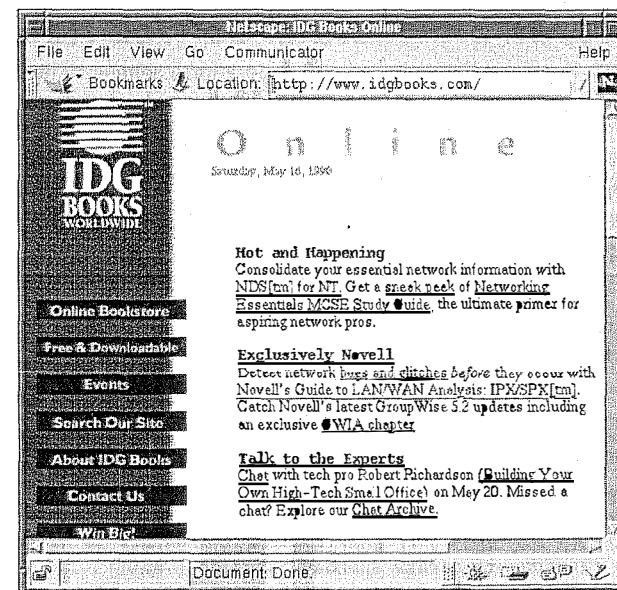
Exemplul cu comenzile HTTP arată rezultatul comenzii GET. Aceasta este cea mai uzuală comandă HTTP, deoarece determină serverul să returneze un document HTML specificat.



Celelalte două comenzi HTTP sunt HEAD și POST. Comanda HEAD este asemănătoare cu GET; ea determină serverul să returneze tot ce este în document cu excepția corpului. Comanda POST este folosită pentru a trimite informații către server; acesta va decide cum va reacționa la primirea ei.

Figura 20-3

Pagina de bază a sitului www.idgbooks.com, afișată cu browserul de Web Netscape Navigator.



Este HTTP un standard Internet?

În ciuda utilizării foarte răspândite în World Wide Web din 1990, până nu de mult HTTP nu a fost un standard Internet. Toate standardele Internet sunt distribuite ca documente RFC (Request for Comments). Primul document RFC în legătură cu HTTP a fost RFC 1945, „Hypertext Transfer Protocol – HTTP/1.0”, T. Berners-Lee, R. Fielding și H. Frystyk, mai 1996. Totuși, RFC 1945 este considerat un document informativ și nu un standard.

RFC 2068, „Hypertext Transfer Protocol – HTTP/1.1”, R. Fielding, J. Gettys, J. Mogul, H.

Frystyk, T. Berners-Lee, ianuarie 1997 este standardul Internet propus pentru HTTP.

Pentru a citi aceste RFC-uri, examinați adresa <http://www.cis.ohio-state.edu/htbin/rfc/rfc-index.html>.

Pentru a învăța mai multe despre HTTP/1.1 și alte standarde Web, folosiți un browser Web pentru a accesa <http://www.w3.org/pub/WWW/Protocols/>.

Navigarea pe rețea

După ce l-ați văzut cum lucrează, sistemul World Wide Web este mai ușor de înțeles. Una dintre metodele cele mai indicate de a învăța despre Web este de a naviga pe rețea cu un browser de Web. Parcurgerea paginilor de Web este plăcută, deoarece o pagină tipică conține atât text, cât și imagini. De asemenea, navigarea are un element surpriză; puteți executa clic pe legături și să ajungeți în pagini de Web neașteptate. Aceste legături sunt cel mai neobișnuit aspect din Web. Puteți începe examinând o pagina care prezintă vremea de astăzi și, un clic mai târziu, puteți citi online ediția din această săptămână a revistei *Time*.

Înainte de a încerca ceva, bineînțeles că aveți nevoie de un browser de Web. (Trebuie să aveți, de asemenea, o conexiune la Internet pentru sistemul dumneavoastră Linux, dar presupun că ați avut deja grijă de acest lucru.)

Descărcarea browserului de Web Netscape

În ianuarie 1998, Netscape Communications Corporation a anunțat că produsul Netscape Communicator este disponibil gratuit (Netscape Communicator este browser de Web integrat, e-mail, cititor de mesaje și generator de pagini Web – browserul de Web este denumit Navigator.) Aceasta înseamnă că puteți descărca, instala și folosi browserul de Web Netscape pe sistemul dumneavoastră fără să plătiți nici o taxă.

Pentru a descărca și instala Netscape Communicator parcurgeți următorii pași:

1. Intrați în sistem ca root, creați directorul `/usr/local/netscape` și intrați în el cu următoarele comenzi:

```
mkdir /usr/local/netscape
cd /usr/local/netscape
```

2. Verificați dacă este stabilită conexiunea la Internet a sistemului dumneavoastră Linux.

3. Tastați următoarea comandă (scrisă cu caractere aldine):

```
ftp ftp2.netscape.com
Connected to ftp.netscape.com
220 ftp25 FTP server (UNIX(r) System V Release 4.0) ready.
Name (ftp2.netscape.com:root): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: (introduceți adresa dumneavoastră de e-mail)
230>Welcome to the Netscape Communications Corporation FTP
```

server.

230-

230-If you have any odd problems, try logging in with a minus sign (-)

230-as the first character of your password. This will turn off a feature 230-that may be confusing your ftp client program .

230-

230-Please send any questions, comments, or problem reports about 230-this server to ftp@netscape.com.

230-

230 Guest login ok, acces restrictions apply.

Remote system type is UNIX.

Using binary mode to transfer files.

4. La promptul `ftp>` tastați următoarea comandă (scrisă cu caractere aldine):

```
ftp> cd /pub/communicator/4.05/shipping/english/unix/linux20
250 CWD command successful
```

5. Tastați comanda `ls` pentru a examina listingul fișierului:

```
ftp> ls
200 PORT command successful.
201 Opening ASCII mode data connection for *ls.
drwxr-xr-x 1 ftp ftp 512 Apr 01 22:39 base-install
drwxr-xr-x 1 ftp ftp 512 Apr 01 22:40
navigator_standalone
drwxr-xr-x 1 ftp ftp 512 May 07 16:11
professional_edition
226 ASCII Transfer complete.
227
```

După cum puteți vedea, sunt disponibile câteva versiuni de browsere Netscape.

6. Pentru a descărca ediția profesională a Netscape Communicator, mai întâi comutați directorul folosind comanda `cd`:

```
ftp> cd professional_edition
```

7. Setati tipul fișierului pe binar cu următoarea comandă:

```
Ftp> binary
200 Type set to I.
```

8. Procurați-vă Netscape Communicator cu următoarea comandă:

```
ftp> mget communicator*
mget communicator-v405-export.x86-unknown-linux2.0.tar.gz? Y
```

Dacă aveți o conexiune SLIP sau PPP printr-un modem, acest pas va consuma timp, deoarece această comandă descarcă 11.314.759 байți (aproximativ 11 MB). (Mărima fișierului poate fi diferită atunci când veți descărca Netscape.) Cu o legătură de 28800 bps, puteți descărca aproximativ 3 KB pe secundă; la această viteză, pentru descărcarea a 11 MB va fi necesară o oră. Cu o conexiune de viteză mai mare, timpul pentru descărcare va scădea proporțional.

9. Tastați `bye` pentru a părăsi `ftp`.

10. Fișierul descărcat este o arhivă comprimată (extensia `.gz` la sfârșitul numelui fișierului indică faptul că fișierul este comprimat). Folosiți următoarea comandă pentru a extrage conținutul arhivei:

```
tar zxvf communicator*.gz
```

Vor apărea câteva mesaje pe măsură ce fișierele sunt extrase din arhivă.

11. Ar trebui să vedeți un fișier numit `README.install` (numele fișierului se poate schimba în versiunile ulterioare ale pachetului de programe Netscape Communicator). Tastați următoarea comandă pentru a citi acel fișier:

```
more README.install
```

12. Urmăți instrucțiunile din fișierul `README.install` pentru a instala Netscape Communicator pe sistemul dumneavoastră Linux. Metoda recomandată de Netscape pentru instalarea programului este de a lansa în execuție un script de interpretor denumit `ns-install` (numele scriptului se poate schimba în versiunile ulterioare ale programului).

Asta-i tot! Acum puteți porni browserul Netscape tastând `netscape` într-o fereastră `xterm`. (Nu uitați că trebuie să porniți X înainte de a lansa în execuție `netscape`, deoarece Netscape folosește X Window System.)

Să aruncăm o privire asupra Netscape Navigator

Netscape Navigator a pornit ca succesor al aplicației Mosaic, browserul de Web original de la National Center for Supercomputing Applications (NCSA) al Universității Illinois din Urbana-Champaign. Unul dintre primii dezvoltatori ai programului Mosaic, Marc Andreessen, a fost viitorul Netscape Navigator. Netscape Navigator a îmbunătățit browserul Mosaic în câteva privințe, cea mai semnificativă fiind modalitatea prin care Netscape Navigator încarcă o pagină Web.

Când o pagină Web conține imagini înglobate, browserul trebuie să descarce fiecare imagine separat. Mosaic afișează o pagină de Web numai după descărcarea integrală a paginii respective. Netscape Navigator, pe de altă parte, începe afișarea paginii imediat ce devin disponibile părți din aceasta.

De asemenea, Netscape Navigator termină de descărcat o pagină mai repede, deoarece stabilește multiple conexiuni cu serverul Web pentru a descărca în paralel componente separate ale paginii. (Acest proces solicită mai mult serverul Web, dar este benefic pentru utilizator.)

Pornirea Netscape

Pentru a lansa în execuție Netscape, tastați `netscape` în linia de comandă dintr-o fereastră `xterm` (trebuie să porniți X înainte de a lansa în execuție Netscape). Când Netscape pornește, încarcă automat pagina Web identificată de adresa URL `http://home.netscape.com/`.



Secret

Dacă faceți comparație între această sintaxă și aceea a adresei URL descrisă anterior în acest capitol, veți observa că această adresă URL nu pare să aibă un nume de fișier. Când o adresă URL nu are un nume de fișier, serverul Web trimite un fișier HTML prestabilit denumit `index.html` (acesta este numele prestabilit pentru popularele servere Web Apache bazate pe UNIX și NCSA; serverele Web bazate pe Windows NT folosesc un nume de fișier prestabilit diferit).



Observație

Serverele Web conțin mai multe pagini Web, organizate astfel încât puteți porni de la o pagină principală și să „săriți” la alte pagini. Pagina Web principală de pe un server Web se numește pagină de bază (*home page*).

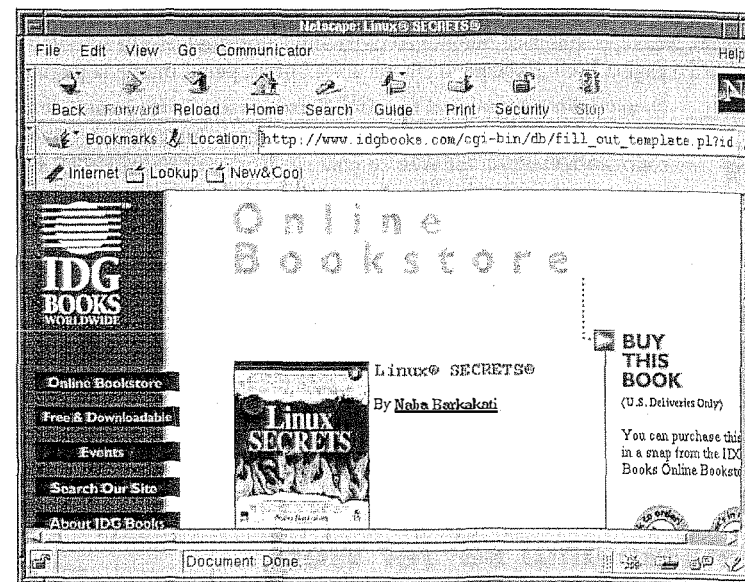
Adresa URL `http://home.netscape.com/` reprezintă pagina de bază a programului Netscape Communications – compania care a dezvoltat Netscape Communicator. Fără o pagină Web, un browser Web nu poate afișa nimic.

Interfața cu utilizatorul a programului Netscape Navigator

Figura 20-4 prezintă o pagină Web din situl Web IDG Books, precum și elementele principale ale ferestrei programului Netscape Navigator.

Figura 20-4

Elementele ferestrei programului Netscape Navigator.



Fereastra Netscape Navigator afișează o interfață Motif cu utilizatorul, cu o bară de titlu ce indică numele documentului curent, precum și o bară de meniu conținând toate meniurile standard (ca de exemplu File și Edit).

Cea mai importantă parte a ferestrei Netscape Navigator este fereastra document – suprafața mare în care Netscape afișează pagina Web, cu textul și imaginile aferente.

Imediat deasupra ferestrei document se află trei articole pe care le puteți activa sau dezactiva din meniul View:

- **Bara cu instrumente de navigație** (Navigation Toolbar) vă oferă acces rapid la câteva articole de meniu uzuale. Butoanele Back și Forward sunt pentru deplasarea între paginile Web pe care le-ați văzut deja; butonul Home vă duce la pagina de bază (care la început este pagina inițială a programului Netscape); butonul Reload (reîncărcare) forțează browserul să descarce din nou pagina curentă; Search (căutare) vă trimite la pagina din care puteți căuta informații pe Internet; butonul Guide (ghid) vă permite să consultați online câteva sfaturi utile; butonul Print (tipărire) tipărește pagina curentă; Security (securitate) prezintă informații despre securitate; butonul Stop vă permite să opriți încărcarea paginii Web curente.

- **Bara cu instrumente de localizare** (Location Toolbar) – afișează locația paginii Web curente sub forma unei adrese URL și permite accesul la semnele dumneavoastră de carte (adrese URL salvate anterior). Puteți, de asemenea, executa clic în această zonă și să introduceți adresa URL a oricărei pagini pe care doriți să o vedeți.

■ **Bara cu instrumente personale** (Personal Toolbar) – oferă acces rapid către articole accesibile prin intermediul butonului Guide din bara cu instrumente de navigație. Aceste butoane vă permit să accesați rapid anumite pagini Web. Butonul Internet, de exemplu, vă afișează o pagină Web în care puteți parcurge informații organizate pe categorii, ca de exemplu afaceri, computere, distracții și știrile principale.

În colțul din dreapta-sus al ferestrei Netscape se află butonul Activity Indicator (marcat cu un *N* mare). Netscape animează butonul Activity Indicator (indicatorul activității) în timp ce descarcă o pagină Web. Executând clic pe acest buton, puteți merge la pagina inițială Netscape.



Secret

În colțul din stânga-jos al ferestrei Netscape se găsește lacătul de securitate. Netscape suportă cu o versiune sigură a protocolului HTTP. Când Netscape se conectează la un server Web ce poate utiliza un protocol HTTP sigur, lacătul de securitate apare încuiat. În caz contrar, lacătul de securitate este deschis, semnificând o conexiune nesigură.

Netscape afișează mesaje de stare pe o suprafață mare din dreapta cheii de securitate. Când Netscape este ocupat cu descărcarea unei pagini Web, afișează procentul de descărcare a documentului.

De-a lungul marginii din dreapta jos a ferestrei, Netscape Navigator afișează câteva pictograme. Executând clic pe acestea, puteți deschide alte ferestre pentru a realiza anumite activități, ca de exemplu citirea mesajelor e-mail sau a unui grup de discuții. Pentru a învăța mai multe despre oricare dintre aceste pictograme, pur și simplu deplasați mouse-ul în dreptul butonului, iar Netscape Navigator afișează un mesaj de ajutor într-o mică fereastră pop-up (și, de asemenea, în zona mesajelor de stare).

Setarea unui server Web

Probabil că știți deja cum să folosiți rețeaua Web, dar probabil ca nu știți cum să setați un server Web astfel încât să puteți oferi tuturor informații prin intermediul paginilor Web. Pentru a deveni un furnizor de informații pe Web, trebuie să activați un server Web pe PC-ul dumneavoastră Linux din Internet. Trebuie, de asemenea, să pregătiți paginile Web, o operație ce se poate dovedi mai pretențioasă decât setarea serverului.



Sugestie

Serverele Web funizează informații folosind protocolul HTTP. Serverele Web mai sunt cunoscute sub numele de daemoni HTTP (deoarece procesele server ce rulează continuu sunt numite *daemoni* în UNIX) sau, pe scurt, HTTPD. Programul serverului Web este denumit, de obicei httpd.

Dintre serverele Web disponibile gratuit, Apache este cel mai popular. Serverul Web Apache a fost la origine o versiune îmbunătățită a serverului HTTPD NCSA, a evoluat rapid ca entitate separată. La fel ca HTTPD NCSA, serverul Apache este dezvoltat și întreținut de o echipă de colaboratori. Apache este disponibil gratuit pe Internet.

Următoarele secțiuni descriu serverul Web Apache.

Instalarea serverului Web Apache

În timpul instalării Linux Red Hat de pe CD-ROM-ul anexat acestei cărți, aveți opțiunea de a instala serverul Web Apache. Așa cum se menționează în capitolul 1, pur și simplu selectați componenta *Mail/WWW/News Tools* când vi se solicită să optați pentru opțiunile ce vor fi instalate. Literele WWW din numele componentei se referă la World Wide Web, care include serverul Apache. Pe măsură ce parcurgeți etapele de instalare descrise în capitolul 1, la un

moment dat trebuie să specificați serviciile care se vor lansa automat atunci când inițializați sistemul dumneavoastră Linux. La acest pas ar trebui să selectați httpd; aceasta asigură pornirea automată a serverului Web Apache.

Parcurgeți următorii pași pentru a verifica dacă programul pentru serverul Web este instalat pe sistemul dumneavoastră:

1. Folosiți comanda `rpm -q apache` pentru a verifica dacă pachetul Apache este instalat:

```
rpm -q apache
apache-1.2.4-4
```

Dacă mesajul emis arată un pachet numit Apache, atunci programul este instalat.

2. Tastați următoarea comandă pentru a verifica dacă procesul httpd (numele programului pentru serverul Web Apache este httpd) rulează:

```
ps ax | grep httpd
```

Mesajul emis ar trebui să conțină un număr de procese httpd. Se obișnuiește rularea mai multor procese pentru servere Web – un proces părinte și câteva procese copil – permițând astfel manevrarea eficientă a mai multor cereri HTTP atribuind fiecare solicitare unui proces httpd.

3. Utilizați programul telnet de pe orice sistem ce are o conexiune de rețea cu PC-ul dumneavoastră Linux și folosiți comanda HTTP HEAD pentru a interoga serverul Web astfel:

```
telnet 192.168.1.1 80 (folositi adresa IP a sistemului dumneavoastra)
```

```
Trying 192.168.1.1...
```

```
Connected to 192.168.1.1.
```

```
Escape character is '^['.
```

```
HEAD / HTTP/1.0
```

```
... Apasati Enter inca o data pentru a introduce o linie goala
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 19 May 1998 11:07:19 GMT
```

```
Server: Apache/1.2.4
```

```
Last-Modified: Thu, 06 Nov 1997 18:20:06 GMT
```

```
ETag: "a014-792-34620a56"
```

```
Content-Length: 1938
```

```
Accept-Ranges: bytes
```

```
Connection: close
```

```
Content-Type: text/html
```

```
Connection closed by foreign host.
```

Dacă obțineți un răspuns similar celui de mai sus, sistemul dumneavoastră are deja instalat și setat corect serverul Web Apache. Nu trebuie decât să înțelegeți configurația pentru a putea plasa documentele HTML în directorul adecvat.

De ce este denumit „Apache“?

Conform informațiilor despre proiectul serverului Web Apache de la adresa <http://www.apache.org/info.html>, grupul Apache s-a format în martie 1995 din persoane ce furnizau fișiere patch ce fuseseră scrise pentru a remedia erorile din NCSA HTTPD 1.3. Rezultatul, după repararea NCSA HTTPD, a fost ceea ce ei au numit *A PaCHy* server; astfel a luat ființă numele Apache.

În conformitate cu sondajul despre servere Web (Netcraft Web Server Survey) din mai 1998, pe

care-l puteți găsi la adresa <http://www.netcraft.co.uk/Survey/>, serverul Web Apache este cel mai popular – 48,26 % din 2.308.502 situri au raportat că folosesc serverul Apache. Microsoft Internet Information Server (IIS) se află pe locul doi, la mare distanță, cu 21,34% din procentul siturilor.



Sugestie

Folosiți un server Web pentru a încărca pagina de bază din sistemul dumneavoastră. Dacă adresa IP a sistemului dumneavoastră este 192.168.1.100, folosiți adresa URL `http://192.168.1.100/` și vedeți ce se întâmplă. Ar trebui să vedeți o pagină Web cu titlul „Pagina de test pentru instalarea serverului Web Apache din Linux Red Hat”.

Configurarea serverului Web Apache

Linux Red Hat configurează programul pentru serverul Web Apache pentru a folosi următoarele directoare:

- Programul pentru serverul Web – `httpd` – este instalat în directorul `/usr/sbin`.
- Fișierele de configurare pentru serverul Web Apache sunt localizate în directorul `/etc/httpd/conf`. Cele trei fișiere de configurare – `access.conf`, `httpd.conf` și `srm.conf` – sunt descrise în următoarele patru secțiuni.
- Serverul Web Apache este setat pentru a deservi documentele HTML din directorul `/home/httpd/html`. Prin urmare, ar trebui să vă plasați paginile dumneavoastră Web în directorul `/home/httpd/html`.
- Dacă aveți vreunul din programele Common Gateway Interface (CGI) – programe ce pot fi invocate de serverul Web pentru a accesa alte fișiere și baze de date – ar trebui să le plasați în directorul `/home/httpd/cgi-bin`.
- Directorul `/var/log/httpd` este destinat pentru fișiere de conectare la serverul Web (fișiere de acces și fișiere de erori).
- Programul script `/etc/rc.d/init.d/httpd` pornește procesul `httpd` când sistemul dumneavoastră Linux se inițializează.

Fișierele de configurare Apache

Operațiunile serverului Apache sunt controlate de trei fișiere de configurare localizate în directorul `/etc/httpd/conf`. Fișierele de configurare controlează cum rulează serverul, ce documente deservește și cine poate accesa aceste documente.

Iată descrierile celor trei fișiere de configurare:

Nume fișier	Scop
<code>httpd.conf</code>	Fișier de configurare pentru serverul Apache. Specifică atributele generale ale serverului, precum numărul portului și directorul în care sunt localizate directoarele serverului.
<code>srm.conf</code>	Fișier de configurare pentru resursele serverului – documente și alte informații pe care serverul Web le furnizează utilizatorilor. De exemplu, fișierul <code>srm.conf</code> specifică unde sunt localizate documentele. (Observați că <code>srm</code> vine de la <i>server resource map</i> – harta cu resursele serverului.)
<code>access.conf</code>	Fișier de configurare ce controlează accesul la serverul Web. Puteți controla accesul la întregul server Web, precum și la anumite directoare.

Următoarele secțiuni furnizează informații cheie despre fiecare fișier de configurare. De obicei, nu trebuie să schimbați nimic în fișierele de configurare pentru a folosi serverul Web Apache. Totuși, este util de știut formatul fișierelor de configurare și semnificația diferitelor cuvinte cheie folosite în acestea.

Pe măsură ce studiați fișierele de configurare din directorul `/etc/httpd/conf`, aveți în vedere următoarele reguli de sintaxă:

- Fiecare fișier de configurare este un fișier text pe care puteți să-l editați cu editorul dumneavoastră favorit și să-l vizualizați cu ajutorul comenzii `more`.
- Toate liniile de comentariu încep cu semnul `#`.
- Fiecare linie poate conține numai o singură directivă.
- Spațiile suplimentare și liniile vide sunt ignorate.
- Toate intrările, cu excepția căilor pentru fișiere și a adreselor URL, sesizează diferența dintre majuscule și literele mici.

Fișierul de configurare `httpd.conf`

Fișierul `/etc/httpd/conf/httpd.conf` este fișierul de configurare al daemonului HTTP fundamental – el controlează modul de rulare al serverului Web Apache. De exemplu, fișierul `httpd.conf` specifică numele portului pe care-l folosește serverul, numele sitului Web și adresa de e-mail la care este trimisă poșta în cazul unor probleme cu serverul.

Iată două articole interesante din fișierul `httpd.conf`:

- **ServerAdmin** este adresa de e-mail pe care serverul Web o furnizează clienților în cazul apariției unor erori. Valoarea prestabilită pentru **ServerAdmin** este `root@localhost` și trebuie setată pe o adresă de e-mail validă astfel încât oricine din Internet s-o poată folosi pentru a raporta erori din situl dumneavoastră Web.
- **ServerName** specifică numele calculatorului gazdă pentru situl dumneavoastră Web (de forma `www.domeniul.dumneavoastra`). Numele ar trebui să fie un nume de domeniu înregistrat, pe care alții îl pot localiza prin intermediul serverului lor de nume. Iată un exemplu:
Server Name `www.myhost.com`

Mult mai multe directive controlează modul în care lucrează serverul Web Apache. Următoarea listă sintetizează câteva din directivele pe care le puteți utiliza în fișierul `httpd.conf`. Pe majoritatea le puteți lăsa la valorile lor prestabilite, dar este importantă cunoașterea acestor directive în cazul în care administrați un server Web.

Server Type *tip* – Specifică modul în care serverul HTTP este executat de Linux. Tipul (*tip*) poate fi **inetd** (pentru a utiliza serverul prin intermediul daemonului `inetd`) sau **autonom** (**standalone**) (pentru a utiliza serverul ca un proces autonom). Pentru performanțe mai bune ar trebui să utilizați serverul autonom.

Port *num* – Specifică faptul că demonul HTTP ar trebui să asculte la portul *num* (un număr între 0 și 65535) pentru solicitări de la clienți. Portul prestabilit pentru HTTPD este 80. Ar trebui să lăsați numărul portului la valoarea lui prestabilită, deoarece clienții consideră că portul HTTP este 80. Dacă serverul dumneavoastră nu folosește portul 80, adresa URL pentru serverul dumneavoastră trebuie să specifice numărul portului.

User *nume* [*#id*] – Specifică numele utilizatorului (sau identificatorul – ID) folosit de daemonul HTTP când rulează în modul autonom. Puteți lăsa această directivă la setarea prestabilită (`nobody`). Dacă specificați identificatorul unui utilizator, folosiți prefixul `diez` (`#`) pentru identificatorul numeric.

Group *nume* [*#id*] – Specifică numele grupului (sau identificatorul) pentru daemonul HTTP când rulează în modul autonom. Numele prestabilit al grupului este `nobody`.

ServerAdmin *webmaster@company.com* – Furnizează serverului adresa e-mail a persoanei care administrează serverul Web. În cazul unor erori, serverul furnizează această adresă astfel încât utilizatorii să poată raporta erorile.

ServerRoot *cale* – Specifică directorul unde este localizat serverul Web. În mod prestabilit, fișierele de configurare și consemnare (log) sunt localizate în subdirectoare ale acestui director.

ServerName *www.company.com* – Setează numele de calculator gazdă al serverului pe valoarea *www.company.com*, înlocuind astfel numele real de calculator gazdă. Nu puteți pur și simplu să inventați un nume; acesta trebuie să fie un nume valid din sistemul de nume al domeniilor (Domain Name System – DNS) a sistemului dumneavoastră

StartServers *num* – Setează numărul proceselor copil ce pornesc imediat ce serverul Web Apache intră în execuție.

MaxSpareServers *num* – Setează numărul maxim dorit de procese server copil inactive (un proces copil este considerat inactiv dacă nu manevrează o cerere HTTP).

Timeout *numsec* – Setează numărul de secunde cât așteaptă serverul pentru ca un client să trimită o interogare după ce acesta stabilește conexiunea. Valoarea prestabilită pentru Timeout este de 300 de secunde (5 minute).

ErrorLog *nume_fisier* – Setează fișierul în care httpd consemnează erorile pe care le întâmpină. Dacă numele fișierului nu începe cu un slash (/), atunci el este considerat relativ la ServerRoot. Fișierul ErrorLog prestabilit este /var/log/httpd/error_log. Întrăririle de eroare din fișier se includ evenimente ca de exemplu reporniri ale serverului și mesaje de avertizare, după cum urmează:

```
[Sun Apr 19 01:02:03 1998] Server configured -- resuming normal
operations
[Wed Apr 22 01:24:29 1998] httpd: caught SIGTERM, shutting down
```

TransferLog *nume_fisier* – Setează fișierul în care httpd înregistrează toate accesările clientului (inclusiv accesările nereușite). Fișierul TransferLog prestabilit este /var/log/httpd/access_log. Următorul exemplu arată cum este înregistrat în fișierul TransferLog un acces tipic:

```
192.168.1.200 - - [19/May/1998:06:22:15 -0400] „GET / HTTP/1.0“
200 1938
192.168.1.200 - - [19/May/1998:06:22:15 -0400] „GET / apache_pb.gif
HTTP/1.0“ 200 2326
```

Prima intrare este pentru textul fișierului; a doua este pentru o imagine înglobată. Ultimele două articole din fiecare linie arată codul de stare returnat de server, urmat de numărul de baiți trimiși de server.

PidFile *nume_fisier* – Setează fișierul în care httpd își stochează identificatorul de proces. Fișierul PidFile prestabilit este /var/run/httpd.pid. Puteți folosi această informație pentru a opri sau reporni daemonul HTTP. Următorul exemplu arată modul de repornire a httpd:

```
kill -HUP cat/var/run/httpd.pid
```

TypesConfig *nume_fisier* – Specifică fișierul ce conține extensiile fișierelor pentru tipurile de date MIME. (MIME vine de la *Multipurpose Internet Mail Extensions*, care definește o modalitate de impachetare a documentelor atașate într-un singur fișier de mesaj.) Serverul raportează aceste tipuri MIME clienților. Dacă nu specificați o directivă TypesConfig, httpd consideră că fișierul TypesConfig este /etc/mime.types. Iată câteva rânduri din fișierul prestabilit /etc/mime.types:

```
(multe linii au fost sterse...)
Text/html          html
Image/ief          ief
Image/jpeg         jpe jpeg jpg
Application/x-latex latex
application/x-troff-man man
```

Fiecare linie indică tipul MIME (ca de exemplu text/html), urmat de extensiile fișierului pentru acel tip (html).

Fișierul de configurare srm.conf

Fișierul de configurare a resurselor, etc/httpd/conf/srm.conf, specifică locația paginilor Web și modul de specificare a tipului de date pentru diferite fișiere. Pentru a porni, puteți lăsa directivele la setările lor prestabilite.

Iată câteva directive de configurare a resurselor pentru serverul Web Apache:

DocumentRoot *cale* – Specifică directorul în care serverul HTTP găsește paginile Web. În Linux Red Hat, valoarea prestabilită pentru DocumentRoot este /home/httpd/html. Dacă plasați documentele dumneavoastră HTML în alt director, atribuiți directivei DocumentRoot numele celui director.

UserDir *nume_director* – Specifică subdirectorul aflat în directorul de bază al utilizatorului, în care serverul HTTP caută paginile Web când apare un nume de utilizator în adresa URL (într-o adresa URL, ca de exemplu http://www.access.digex.net/~naba/, care include un nume de utilizator având prefixul tildă). Valoarea prestabilită pentru UserDir este public_html, care înseamnă că paginile Web ale unui utilizator vor fi în subdirectorul public_html din directorul de bază al celui utilizator. Dacă nu doriți să permiteți utilizatorilor să aibă pagini Web, specificați disabled ca nume al directorului în directiva UserDir.

DirectoryIndex *nume_fisier* – Indică fișierul prestabilit ce va fi returnat de server când clientul nu specifică nici un document. Valoarea prestabilită pentru DirectoryIndex este index.html. Dacă httpd nu găsește acest fișier, returnează un index (de obicei, o listă de fișiere cu aspect ordonat) al celui director.

AccessFileName *nume_fisier* – Specifică numele fișierului ce poate apărea în fiecare director conținând documente și indică cine are permisiunea să acceseze conținutul celui director. Valoarea prestabilită pentru AccessFileName este .htaccess. Sintaxa acestui fișier este la fel cu cea a fișierului access.conf, discutat în secțiunea următoare.

AddType *tip/subtip extensie* – Asociază o extensie de fișier cu un tip de date MIME (de forma *tip/subtip*, ca de exemplu text/plain sau image/gif). Astfel, dacă doriți ca serverul să rateze fișierele cu extensia .lst ca fișiere text simple, trebuie să faceți următoarea specificație:

```
AddType text/plain lst
```

Tipurile MIME prestabilite și extensiile sunt listate în fișierul /etc/mime.types.

AddEncoding *type extensie* – Asociază un tip de codificare cu o extensie de fișier. Dacă doriți ca serverul să marcheze fișierele cu extensia .gz ca fișiere codificate cu metoda de codificare x-gzip (numele standard pentru codarea GZIP), trebuie să specificați următoarele:

```
AddEncoding x-gzip gz
```

DefaultType *tip/subtip* – specifică tipul MIME pe care serverul ar trebui să-l folosească în cazul în care nu-l poate determina după extensia fișierului. Dacă nu specificați

`DefaultType`, `httpd` consideră că tipul MIME este `text/html`. În fișierul prestabilit `srn.conf` existent pe CD-ROM-ul anexat acestei cărți, `DefaultType` este specificat ca `text/plain`.

`Redirect requested-file actual-URL` – Specifică faptul ca orice cerere pentru `requested-file` să fie redirecționată către `actual-URL`.

`Alias requested-dir actual-dir` – Specifică faptul că serverul folosește `actual-dir` pentru a localiza fișiere în directorul `requested-dir` (cu alte cuvinte, `requested-dir` este un pseudonim pentru `actual-dir`). Dacă doriți ca cererile pentru directorul `/icons` să meargă către `/home/httpd/icons`, trebuie să specificați următoarele:

`Alias/icons/ /home/httpd/icons/`

`ScriptAlias requested-dir actual-dir` – Specifică numele real al directorului în care sunt localizate fișierele script Common Gateway Interface (CGI). Fișierul prestabilit `srn.conf` conține următoarea directivă:

`ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/`

Această directivă înseamnă că atunci când un browser Web solicită un script, ca de exemplu `/cgi-bin/test-cgi`, serverul HTTP lansează în execuție script-ul `/home/httpd/cgi-bin/test-cgi`.

`FancyIndexing on [off]` – Activează sau dezactivează afișarea ordonată a listelor de directoare, cu pictograme și mărimile fișierelor.

`DefaultIcon iconfile` – Specifică locația pictogramei prestabilite pe care serverul trebuie s-o folosească pentru fișiere fără informații despre pictogramă. În mod prestabilit, `DefaultIcon` este `/icons/unknown.gif`.

`ReadmeName nume_fisier` – Specifică numele unui fișier README al cărui conținut este adăugat la sfârșitul unei liste de directoare generate automat. Valoarea prestabilită pentru `ReadmeName` este README.

`HeaderName nume_fisier` – Specifică numele unui fișier antet al cărui conținut este inserat înaintea unei liste de directoare generate automat. Valoarea prestabilită pentru `HeaderName` este HEADER.

`AddDescription „file description” nume_fisier` – Specifică faptul că șirul *file description* să fie afișat lângă argumentul `nume_fisier` specificat în lista de directoare. Puteți folosi un caracter de înlocuire, ca de exemplu `*.html`, ca `nume_fisier`.

`AddIcon iconfile extension1 extension2 ...` – Asociază o pictogramă cu una sau mai multe extensii de fișiere. Următoarea directivă asociază fișierul pictogramă `/icons/movie.xbm` cu extensiile `.mpeg` și `.qt`:

`AddIcon /icons/text.gif .txt`

`AddIconByType iconfile MIME-types` – Asociază o pictogramă cu un grup de tipuri de fișiere specificate sub forma unor caractere de înlocuire a tipurilor MIME (ca de exemplu `text/* sau image/*`). Pentru a afișa fișierul pictogramă `/icons/text.xbm` cu toate tipurile text, specificați următoarele:

`AddIconByType (TXT,/icons/text.gif) text/*`

Această directivă comunică, de asemenea, serverului să folosească TXT în locul pictogramelor pentru clienții care nu pot accepta imagini. (Browsele comunică serverului ce tipuri de date pot accepta.)

`AddIconByEncoding iconfile encoding1 encoding2 ...` – Specifică o pictogramă ce va fi afișată pentru unul sau mai multe tipuri de codificare (ca de exemplu `x-compress` sau `x-tar`).

`IndexIgnore filename1 filename2 ...` – Instruiește serverul să ignore numele de fișiere specificate (ele conțin de obicei caractere de înlocuire) când pregătește o listă de directoare. Pentru a nu afișa în listă fișierele README, HEADER și toate fișierele al căror nume încep cu punct (`.`), trebuie să specificați următoarele:

`IndexIgnore */.??* *~ *# */HEADER* */README* */RCS`

`IndexOptions option1 option2 ...` – Indică opțiunile pe care le doriți în lista de directoare pregătită de server. Opțiunile pot include una sau mai multe din următoarele:

- `FancyIndexing` activează afișarea ordonată a directoarelor.
- `IconsAreLinks` determină pictogramele să se comporte ca legături.
- `ScanHTMLTitles` arată o descriere a fișierelor HTML.
- `SuppressLastModified` oprește afișarea datei ultimei modificări.
- `SuppressSize` oprește afișarea mărimii fișierelor.
- `SuppressDescription` oprește afișarea descrierii fișierelor.

`ErrorDocument errortype filename` – Specifică un fișier pe care serverul ar trebui să-l trimită când survine o eroare de un anumit tip. Dacă nu aveți această directivă, serverul trimite un mesaj de eroare înglobat. `Errortype` poate lua una din următoarele valori:

- 302 – REDIRECT
- 400 – BAD_REQUEST
- 401 – AUTH_REQUIRED
- 403 – FORBIDDEN
- 404 – NOT_FOUND
- 500 – SERVER_ERROR
- 501 – NOT_IMPLEMENTED

Fișierul de configurare `access.conf`

Fișierul `/etc/httpd/conf/access.conf` vă permite să controlați cine poate accesa anumite directoare din sistem. Acesta este fișierul de configurare pentru acces global. În fiecare director, puteți avea alt fișier de configurare pentru acces, cu numele specificat de directiva `AccessFileName` din fișierul `srn.conf`. (Acel fișier de configurare pentru accesul într-un director are numele prestabilit `.htaccess`.)

Eliminând majoritatea liniilor de comentariu, fișierul `access.conf` are următorul format:

```
# Urmatorul nume de director ar trebui
# sa fie identic cu cel specificat de directiva DocumentRoot din fisierul
# srm.conf
<Directory /home/httpd/html>
    Options Indexes FollowSymLinks
    AllowOverride None
    order allow,deny
    allow from all
</Directory>
```

```
# Directorul ar trebui sa aiba aceeasi
# locatie ca directorul cgi-bin
```

```
< Directory /home/httpd/cgi-bin>
AllowOverride None
Options ExecCGI
</Directory>
```

Observați că fișierul `access.conf` folosește o sintaxă diferită față de fișierele `httpd.conf` și `srm.conf`. Dacă știți limbajul HTML, puteți vedea că sintaxa este similară cu a acestuia. Diferite directive pentru controlul accesului sunt incluse între paranteze unghiulare, ca de exemplu `<Directory> ... </Directory>`.

Următoarea listă descrie câteva dintre directoarele de control al accesului. În particular, observați directiva `AuthUserFile`; puteți avea controlul accesului pe bază de parolă pentru anumite directoare.

`Options opt1 opt2 ...` – Specifică opțiunile pentru controlul accesului pentru secțiunea de directoare în care apare această directivă. Opțiunile pot fi una sau mai multe dintre următoarele:

- `None` dezactivează toate facilitățile de control al accesului.
- `All` activează toate facilitățile pentru director.
- `FollowSymLinks` activează serverul pentru a urmări legăturile simbolice.
- `SymLinksIfOwnerMatch` urmărește legături simbolice numai dacă directorul legat este al aceluiași utilizator ca și acest director.
- `ExecCGI` permite executarea script-urilor CGI din director.
- `Includes` permite serverului să includă fișiere în acest director.
- `Indexes` permite clienților să solicite indexuri (liste de directoare) pentru acest director.
- `IncludesNoExec` dezactivează caracteristica `exec`.

`AllowOverride directive1 directive2 ...` – Specifică directivele de control al accesului pentru directoare care pot fi suprascrise. Lista de directive poate conține una sau mai multe dintre următoarele:

- `None` oprește suprascrierea oricărei directive.
- `All` oprește suprascrierea oricărei directive referitoare la un director.
- `Options` permite folosirea directivei `Options` în fișierul de la nivelul directorului.
- `FileInfo` permite folosirea directivelor de codificare `AddType` și `AddEncoding`
- `AuthConfig` permite folosirea directivelor `AuthName`, `AuthType`, `AuthUserFile` și `AuthGroupFile`.
- `Limit` permite folosirea directivelor `Limit` în fișierul de configurare pentru acces în directoare.

`AuthName nume` – Specifică numele autorizat pentru un director.

`AuthType tip` – Specifică tipul de autorizare ce va fi folosit. Singurul tip de autorizare permis este `Basic`.

`AuthUserFile filename` – Specifică fișierul în care sunt stocate numele utilizatorilor și parolele pentru autorizare. Următoarea directivă setează pentru autorizare fișierul `/etc/httpd/conf/.htpasswd`:

`AuthUserFile /etc/httpd/conf/.htpasswd`

Trebuie să creați fișierul de autorizare cu programul de asistență `/usr/sbin/htpasswd`, localizat în directorul `/etc/httpd/support`. Pentru a crea fișierul de autorizare și a adăuga parola pentru un utilizator numit `jdoe`, trebuie să specificați următoarele:

```
/usr/sbin/htpasswd -c /etc/httpd/conf/.htpasswd jdoe
```

Adding password for jdoe.

New password: (introduceți parola)

Re-type new password: (introduceți din nou aceeași parola)

`AuthGroupFile nume_fisier` – Specifică fișierul ce trebuie consultat pentru autentificare de către o listă de grupuri de utilizatori.

`order ord` – Această directivă specifică ordinea în care alte două directive – `allow` și `deny` – sunt evaluate. Ordinea poate fi una din următoarele:

- `deny, allow` evaluează directiva `deny` înaintea directivei `allow`
- `allow, deny` – evaluează directiva `allow` înaintea directivei `deny`
- `mutual-failure` – permite accesul numai acelor calculatoare gazdă care sunt în lista `allow`.

`deny from host1 host2 ...` Această directivă specifică calculatoarele gazdă care au accesul interzis.

`allow from host1 host2 ...` Această directivă specifică calculatoarele gazdă care au accesul permis. Dacă doriți ca toate calculatoarele gazdă dintr-un anumit domeniu să acceseze documentele Web dintr-un director, trebuie să specificați următoarele:

```
order deny, allow
allow from .nws.noaa.gov
```

`require entity en1 en2 ...` – Această directivă specifică utilizatorii care pot accesa un director. Entity poate fi una din următoarele:

- `user` permite o listă numai cu anumiți utilizatori.
- `group` permite o listă numai cu anumite grupuri.
- `valid-user` permite tuturor utilizatorilor listați în fișierul `AuthUserFile` accesul la director (cu condiția să introducă parola corectă).

Rezumat

Sistemul World Wide Web (WWW sau Web) a propulsat Internetul în prim-plan, deoarece browserele Web facilitează parcurgerea de către utilizatori a documentelor Web stocate pe diferite calculatoare gazdă din Internet. Dacă administrați calculatoare și rețele pentru o companie sau conduceți o mică afacere, este foarte probabil că va trebui să setați și să administrați un server Web. Datorită facilităților de rețea înglobate, un PC Linux poate deveni cu ușurință server World Wide Web. Acest capitol descrie cum să setați și să configurați un server Web pe un PC Linux.

Din lectura acestui capitol, ați învățat următoarele:

- World Wide Web este posibil datorită existenței unui format standard pentru documente și a unui protocol standard pentru transferarea acestora prin rețea. Formatul pentru documente se numește Hypertext Markup Language sau HTML. Protocolul standard pentru schimbul de documente se numește Hypertext Transfer Protocol sau HTTP.

- ▶ Sistemul Web are o arhitectură client-server, serverele Web furnizând documentele HTML (deseori denumite *pagini Web*) browserelor Web, care sunt clienții.
- ▶ Pentru a identifica în mod unic pagini Web și alte resurse de rețea, se folosește sintaxa Uniform Resource Locator (URL). O adresă URL identifică locația documentului (numele calculatorului și directorul), precum și protocolul folosit pentru transferarea documentului (ca de exemplu http sau ftp).
- ▶ Acest capitol vă prezintă modul de descărcare și instalare a aplicației Netscape Navigator. Ar trebui să încercați un browser înainte de a seta un server Web.
- ▶ Cel mai popular server Web este Apache. Puteți instala serverul Web Apache în timpul instalării Linux Red Hat. Acest capitol descrie fișierele de configurare pentru serverul Web Apache.

Capitolul 21

Administrarea unei afaceri cu Linux

În acest capitol

- ▶ Veți înțelege rolul pe care Linux îl poate avea în afaceri
- ▶ Veți examina operațiile cele mai adecvate sistemului Linux
- ▶ Veți utiliza un PC Linux ca server LAN Manager
- ▶ Veți tipări din Linux pe o imprimantă LAN Manager
- ▶ Veți examina posibilitățile de afaceri cu Linux
- ▶ Veți utiliza Linux în anumite domenii

Deoarece Linux este disponibil gratuit, mulți cred că nu este adecvat pentru o afacere. Câteva din motivele cel mai des invocate sunt lipsa suportului tehnic și lipsa aplicațiilor pentru afaceri. Acestea pot fi motive serioase pentru a evita folosirea Linux în conducerea unor afaceri, cu toate că multe afaceri mici pot fi conduse în întregime cu ajutorul Linux. Prin conducerea unei afaceri mă refer la toate treburile dumneavoastră zilnice, ca de exemplu actualizarea registrelor și redactarea corespondenței de afaceri.



Observație

Comaniile mai mici ai căror angajați au experiență în UNIX, de exemplu, se pot descurca pentru a conduce o afacere cu Linux. Dacă sunteți un consultant ce oferă un sistem complet, la cheie, pentru o aplicație comercială, puteți folosi Linux, păstrând astfel costul destul de scăzut pentru a fi în avantaj față de concurență. Puteți chiar să luați în considerare furnizarea suportului necesar pentru a vă asigura clienții că se pot baza pe soluția pe care o furnizați.

Situația s-a schimbat și în alte privințe. O companie numită Caldera Inc., vinde un Linux comercial numit OpenLinux, care conține o suprafață de lucru grafică, aplicații de productivitate, incluzând WordPerfect pentru Linux și StarOffice, server și browser Netscape Internet și facilitatea de a accesa servere NetWare. Cu siguranță, toate aceste facilități vă costă ceva bani, dar costul este totuși mai mic decât al unui pachet similar care nu folosește Linux. Compania Caldera Inc. a fost înființată de Ray Noorda de la cunoscuta companie Novell.

Nu voi descrie produsele Caldera în acest capitol; puteți afla mai multe la adresa URL <http://www.caldera.com>. În schimb, mă voi concentra asupra activităților care în mod inevitabil au personalul tehnic care se poate descurca bine cu un sistem de operare care se poate întreține singur, precum Linux. De fapt, multe dintre discuțiile din acest capitol ar trebui să vă dea idei despre noi tipuri de afaceri – bazate pe popularitatea actuală a Internetului și a sistemului World Wide Web.

Strategia acestui capitol

Acest capitol nu descrie un anumit subiect tehnic. Totuși, conducerea unei afaceri are multe aspecte, iar în multe domenii de activitate există multiple metode de a executa o anumită operație. Cu toate

că acest capitol se concentrează asupra domeniilor ce pot beneficia de facilitățile Linux, puteți folosi totuși instrumente diferite pentru fiecare activitate. Ca factor de decizie, dumneavoastră trebuie să hotărâți care este cea mai bună soluție pentru afacerea dumneavoastră.

Pentru a face informațiile din acest capitol mai folositoare în procesul de luare a deciziilor, folosesc următoarea strategie:

- **Rolul sistemului Linux într-o afacere.** Voi descrie, pe scurt, tipurile de cerințe ale unei afaceri pentru care Linux este adecvat și pe cele pentru care Linux este mai puțin recomandat sub aspectul costurilor, sau pur și simplu nu este indicat.
- **Operații specifice pentru Linux.** Voi discuta câteva întrebări specifice ale sistemului Linux. Puteți considera aceste utilizări ca fiind meniul din care alegeți modul de utilizare a sistemului Linux în afacerea dumneavoastră
- **Linux în anumite domenii.** Mă voi concentra asupra câtorva afaceri specifice, pentru care Linux poate lucra deosebit de bine.



Observație

Cu informația prezentată în acest fel, ar trebui să puteți decide partea din afacere și componentele care pot fi administrate folosind Linux.

Rolul sistemului Linux într-o afacere

Există multe tipuri de afaceri, iar pentru o anumită afacere sunt multe aspecte care pot beneficia de avantajele PC-urilor Linux. Dacă vă gândiți serios să folosiți Linux în afacerea dumneavoastră, gândiți-vă cu atenție la componentele pentru care îl puteți folosi cel mai bine. Cele mai multe afaceri, de exemplu, au nevoie de aplicații eficiente: procesoare de texte, foi de calcul tabelar, etc. Inițial, Linux era deficitar la acest capitol, dar acum utilizatorii lui pot alege dintre câteva aplicații eficiente de birou, ca de exemplu WordPerfect pentru Linux (<http://www.sdcorp.com>), Applixware Office Suite (<http://www.redhat.com/products/>) și StarOffice Suite (<http://www.caldera.com/ivp/onlinecatlog/staroffice.html>). Aceste produse costă ceva bani, dar costul este comparabil cu cel al pachetului cu aplicații de birou al firmei Microsoft.

Pe de altă parte, în ceea ce privește TCP/IP sau dezvoltarea aplicațiilor UNIX, Linux furnizează tot programul necesar, fără costuri suplimentare. Evident, este eficient să folosiți Linux ca platformă pentru lucrul în rețea, pentru dezvoltare de aplicații software sau ca server Web.

O caracteristică atractivă a sistemului de operare Linux este aceea că poate rula pe același PC pe care-l puteți folosi pentru Windows 95 sau OS/2. Cu alte cuvinte, puteți standardiza afacerea dumneavoastră pe o arhitectură hardware puternică la un preț redus. Apoi, alegeți ca sistem de operare Linux și alte sisteme populare de care veți avea nevoie în rețea (ca de exemplu Windows 95, Windows NT sau OS/2). Selecția programelor rulate pe un anumit PC depinde de modul în care calculatorul respectiv este utilizat.



Observație

Linux nu este recomandat pentru orice tip de afacere și nici nu este potrivit pentru toate aspectele unei afaceri. Considerați Linux ca un instrument pentru afacerea dumneavoastră, dar nu poate fi și singurul. Exact la fel cum o singură unealtă nu poate fi bună la toate, Linux nu poate rezolva toate problemele dintr-o afacere. Chiar dacă sunteți un susținător al Linux, ar fi bine să nu încercați să-l folosiți ca unic sistem de operare pentru întreaga dumneavoastră afacere. Amintiți-vă zicala „omul potrivit la locul potrivit”. Selecția Linux pentru activitățile pe care le poate executa în mod adecvat.

Ce oferă Linux

Pentru a decide necesitățile pe care le poate satisface Linux în cadrul afacerii dumneavoastră, trebuie să știți exact ce anume oferă acesta. Când vorbesc despre Linux, mă refer la un software Linux tipic, ca de exemplu cel care se află pe CD-ROM-ul anexat acestei cărți. Acesta conține mult mai mult decât sistemul de operare de bază: programe utilitare, aplicații, sisteme pentru creșterea performanței, precum XFree86, X Window System compilatoare și interpretoare pentru dezvoltarea aplicațiilor software.

Versiunea Linux Red Hat conține următoarele componente importante:

- Nucleul Linux 2.0.32, cu facilități înglobate TCP/IP și pentru o gamă variată de hardware, ca de exemplu controllere de disc, drivere CD-ROM, plăci Ethernet și de sunet.
 - Servicii Internet, ca de exemplu protocoalele de rețea SLIP/PPP, serverul Web Apache, poșta electronică (BSD sendmail, elm, pine), mesaje (tin, trn și inn), telnet și ftp.
 - Aplicația software Samba LAN Manager pentru lucrul în rețea al PC-urilor.
 - Sistemul XFree86 3.3.1 X Window System pentru interfața grafică utilizator și aplicații grafice.
 - Executable and Linking Format – ELF pentru fișiere binare și biblioteci partajate (important deoarece ELF este formatul binar folosit de sistemele Sun Solaris și UNIX System V versiunea 4).
 - Compilatoare și interpretoare pentru multe limbaje de programare: compilatoare C și C++ GNU gcc 2.7.2, BASIC, Perl 5.004, limbajele script Tcl 8.0 și Tk 8.0, Berkeley Yacc, generatoarele de analiză GNU bison și generatorul de analiză lexicală flex 2.5.4.a.
 - Modulul Intel Binary Compatibility Specification (iBCS), care permite fișierelor binare din alte sisteme de operare să ruleze pe Linux. În prezent, puteți folosi fișiere binare în Linux de pe următoarele sisteme: 4 UNIX System V Release (ca de exemplu Interactive, Unixware și Dell UNIX), orice sistem 3 UNIX System V Release, SCO UNIX, Xenix V/386 și Xenix 286.
 - O mulțime de aplicații și programe utilitare, inclusiv editorul GNU Emacs, jocuri și programele de comunicații seriale seyon și minicom.
 - Procesoare de texte și programe pentru tipărire, ca de exemplu TeX, LaTeX și groff.
- După cum puteți vedea din această listă, Linux excelează în două zone largi de tehnologie:
- TCP/IP și servicii Internet, inclusiv World Wide Web
 - Dezvoltarea de aplicații în sistemul X și UNIX

Aceste două zone au o temă comună: oamenii care lucrează în aceste domenii sunt experți în calculatoare. Astfel, ei nu-și fac griji în privința aparentei lipse a suportului tehnic pentru Linux. În următoarea secțiune, veți vedea de ce folosesc termenul *aparent* pentru a descrie lipsa oricărei surse de suport tehnic pentru Linux.

Ce-i lipsește (în aparență) sistemului Linux

Nu am scris această secțiune pentru a mă plânge de neajunsurile Linux. La urma urmei, puteți alege orice sistem de operare sau produs (fie el comercial sau freeware) și veți găsi neajunsuri reale sau aparente în fiecare din acestea. Scopul meu este să subliniez neajunsurile pe care le-am observat la Linux și să vă explic cum pot deveni acestea oportunități de afaceri pentru dumneavoastră, care aveți cunoștințele tehnice necesare pentru a instala și configura Linux.

Lipsa aplicațiilor personale de uz curent

Când mă refer la folosirea Linux într-o afacere, primul lucru la care mă gândesc este că Linux nu are aplicații personale de uz curent, precum cele din Microsoft Office. (Microsoft Office este o colecție ce cuprinde câteva aplicații: Microsoft Word pentru procesare de texte, Microsoft Excel pentru foi de calcul tabelar, Microsoft PowerPoint pentru prezentări grafice și Microsoft Access pentru baze de date.)

Până nu demult, acest lucru era adevărat. Acum însă, așa cum am menționat anterior în acest capitol, puteți alege din câteva pachete de birou pentru Linux. Singura problemă ar fi că pachetele cu aplicații de birou pentru Linux nu sunt atât de populare ca și cele pentru Windows.

Totuși, eu nu consider aceasta ca fiind o problemă. Dacă respectați filosofia selectării instrumentului potrivit pentru activitatea potrivită, puteți selecta cele mai răspândite aplicații Windows pentru procesare de texte și lucrul cu foi de calcul tabelar.

Lipsa suportului tehnic

Alt neajuns al sistemului Linux, deseori invocat, este că necesită un nivel de cunoștințe destul de ridicat în domeniul calculatoarelor pentru instalare și configurare. Eu sunt de părere că toate sistemele de operare UNIX necesită instalare și întreținere de către o persoană avizată. În această privință, Linux nu diferă de sistemele UNIX.

Adevărata problemă este că nu există nici o sursă de suport tehnic pentru Linux. Nu puteți da telefon pentru a discuta cu cineva despre vreo dificultate legată de Linux. În multe domenii se preferă un produs comercial furnizat cu suport tehnic.

În realitate, suportul tehnic pentru un produs comercial înseamnă deseori privilegiul de a fi pus „în așteptare” oricând sunați. Apoi, după ce explicați problema, s-ar putea să primiți doar o promisiune că cineva vă va telefona. Cu toate acestea, multe afaceri nu folosesc Linux datorită lipsei suportului tehnic.

Consider că această situație este mai degrabă o problemă aparentă decât una reală, deoarece grupurile de discuții Linux din Internet au de obicei răspunsuri la orice întrebare a dumneavoastră. Puteți începe prin a citi cele mai frecvente întrebări (Frequently Asked Questions – FAQ) pentru grupuri de discuții Linux. Dacă întrebarea dumneavoastră nu a fost pusă încă, puteți pur și simplu să afișați un articol de mesaj, care să explice problema dumneavoastră. De obicei veți primi răspuns cam într-o zi – probabil tot atât timp durează să primiți un răspuns și în cazul unor aplicații comerciale ce au suport tehnic.

Mai mult decât atât, această nevoie de suport tehnic poate deveni baza propriei dumneavoastră afaceri. Puteți deveni un consultant ce furnizează suport tehnic pentru instalarea, setarea și configurarea Linux. Depinde de dumneavoastră ce doriți să oferiți prin afacerea dumneavoastră. Poate, de exemplu, doriți să vindeți stații de lucru Linux preconfigurate, cu câteva niveluri de suport tehnic.

Sarcini specifice pentru Linux

Linux excelează în multe sarcini specifice, care se aplică diferitelor tipuri de afaceri. Următoarele secțiuni descriu pe scurt cum puteți folosi Linux în următoarele scopuri:

■ **Stații de lucru UNIX personale.** Dacă organizația dumneavoastră folosește stații de lucru UNIX și nu vă puteți permite stații de lucru Sun sau HP pentru fiecare utilizator, dotați-vă cu PC-uri Linux. Ele sunt accesibile ca preț și sunt sisteme UNIX puternice.

■ **Server pentru un grup de lucru.** PC-urile Linux pot fi configurate ușor pentru partajarea fișierelor și a imprimantei.

■ **Calculator gazdă în Internet.** Linux însuși își are originile în Internet și facilitățile lui de lucru în rețea îl fac un calculator gazdă ideal în Internet. Puteți seta și folosi cu ușurință un PC Linux ca server în Internet cu servicii ca poșta, mesaje, *ftp* și *telnet*.

■ **Server World Wide Web.** Linux Red Hat este dotat cu serverul Web Apache. Puteți avea un server Web în stare de funcționare pe sistemul dumneavoastră imediat după ce terminați instalarea Linux. Trebuie doar să optați pentru instalarea serverului Web (vezi capitolul 1 pentru informații despre selectarea diferitelor pachete pe care le doriți instalate în timpul instalării Linux).

■ **Server/client LAN Manager.** Cu Samba (care este inclus pe CD-ROM-ul anexat cărții), puteți seta un PC Linux ca server LAN Manager. Samba facilitează folosirea Linux într-o afacere ce se bazează pe LAN Manager pentru partajarea fișierelor și a imprimantei. Invers, dacă aveți deja un server LAN Manager, puteți determina PC-ul Linux (cu ajutorul programului Samba) să devină client.

■ **Stație de lucru pentru programatori.** Un PC Linux este o stație de lucru ideală pentru programatorii în UNIX și X. Puteți folosi orice limbaj doriți, de la C și C++ până la Tcl/Tk.

Server pentru un grup de lucru

Un PC 486 sau Pentium configurat cu Linux 2.0.34 (conținut pe CD-ROM-ul anexat acestei cărți) devine un server foarte capabil pentru un grup de lucru. Prin *grup de lucru* se înțelege o mică rețea locală formată din aproximativ 10-20 de calculatoare. Dacă pe toate PC-urile rulează Linux, pe unul din acestea îl puteți configura să fie server pentru fișiere și imprimantă, iar celelalte PC-uri Linux să fie clienți. Partajarea fișierelor se poate face prin Network File Sharing – NFS (partajarea fișierelor în rețea) care este o facilități internă din Linux.

Partajarea fișierelor prin intermediul NFS este simplă, implicând două etape de bază:

- Pe serverul Linux exportați unul sau mai multe directoare, listându-le în fișierul `/etc/exports` (fișierul `man exports` vă arată sintaxa).
- Pe fiecare client PC Linux instalați directoarele exportate de către server. Folosiți comanda `mount` pentru a face acest lucru.



Referință
incrucisată

Puteți activa partajarea tipăririi prin configurarea unei imprimante distante pe fiecare PC Linux client și prin setarea imprimantei fizice ca imprimantă locală a serverului. Capitolul 14 descrie cum trebuie setate imprimantele locale și cele distante în Linux.

Dacă în rețeaua dumneavoastră locală sunt PC-uri pe care rulează Windows for Workgroups, de exemplu, trebuie să configurați PC-ul Linux ca server LAN Manager. Mai târziu, secțiunea „Server LAN Manager” din acest capitol descrie pe scurt programele necesare pentru a configura un PC Linux să lucreze cu LAN Manager.



Sugestie

În loc de a folosi pur și simplu un server Linux pentru un grup de lucru în afacerea dumneavoastră, puteți porni o afacere prin care să vindeți stații de lucru Linux preconfigurate. Ideea de bază este să configurați PC-ul Linux cu programele și componentele hardware adecvate (sistemul de operare Linux, XFree86 și orice program necesar pentru rețele) și să-l vindeți cu service la sediul clientului asigurat.

Deoarece Linux este disponibil gratuit, ideea de a vinde o stație de lucru Linux gata de lucru nu mai este o noutate; prin urmare, vă puteți aștepta la ceva concurență în acest domeniu. Vă

puteți construi afacerea în zona dumneavoastră, deoarece aveți avantajul de a acorda service personal. Bineînțeles, puteți furniza asistență unor companii sau indivizi care rulează Linux pe PC-uri cumpărate în altă parte.

Calculator gazdă în Internet

Capitolul 19 descrie cum puteți configura un PC Linux ca sistem gazdă în Internet. Cu o conexiune de rețea adecvată la un furnizor de servicii Internet (Internet Service Provider), un astfel de PC Linux poate asigura afacerii dumneavoastră prezența pe Internet.

Un calculator gazdă din Internet realizează un ansamblu de operații. Aceste sarcini includ cel puțin următoarele:

- **Poșta electronică**, astfel încât utilizatorii pot trimite și primi mesaje
- **Mesaje**, astfel încât utilizatorii pot citi și participa la diferite grupuri de discuții.
- **Server World Wide Web**, pentru a furniza pagini Web altor utilizatori din Internet.
- **Ftp anonim**, astfel încât alți utilizatori pot descărca informațiile pe care le furnizați (de exemplu, remedii pentru aplicații software și informații tehnice despre produsele dumneavoastră)

Varianta Linux de pe CD-ROM-ul anexat acestei cărți conține toate programele necesare pentru setarea acestor servicii Internet. Pentru cea mai mare parte dintre ele, este necesar un minim de configurare pentru a le aduce în stare de funcționare.



Secret

Motivul principal pentru a selecta Linux ca sistem gazdă în Internet este avantajul preț-calitate pe care-l are Linux față de alte sisteme UNIX. Linux rulează pe PC-uri obișnuite – o piață în care competiția este foarte aprigă, iar sistemele Pentium foarte puternice costă mult mai puțin decât stațiile de lucru UNIX similare. Instalați o versiune Linux (ca de exemplu cea de pe CD-ROM-ul anexat acestei cărți) și PC-ul Pentium va fi transformat într-o puternică stație de lucru UNIX, comparabilă cu cele ale unor producători precum Sun, HP sau IBM. Din experiență vă pot spune că un PC Linux are mai multe facilități interne decât o stație de lucru comercială UNIX, pentru care multe facilități sunt disponibile în schimbul unei sume suplimentare.

Concluzia este că un PC Linux configurat complet costă mult mai puțin decât o stație de lucru comercială UNIX configurată similar, PC-ul Linux oferind performanțe comparabile cu acelea ale stațiilor de lucru UNIX. De aceea, un PC Linux este atractiv ca sistem gazdă în Internet.

Pe lângă avantajul preț-performanță, un PC Linux poate fi configurat ca *parafoc* – un sistem care izolează două rețele, permițând numai pachetelor de date selectate să treacă dintr-o rețea în alta. Când conectați o rețea locală (LAN) la Internet, puteți folosi un parafoc pentru a o izola de restul Internetului.

Server World Wide Web

Un PC Linux conectat la Internet este serverul Web ideal pentru afacerea dumneavoastră. Capitolul 20 vă oferă informații detaliate despre modul de setare al serverului Web. De fapt, operația se reduce la setarea fișierelor de configurare ale serverului Apache și inițierea utilizării acestuia. Linux Red Hat conține deja serverul Web Apache. (Sarcina mai grea este realizarea paginilor Web – conținutul pentru World Wide Web.)

În prezent, majoritatea firmelor sunt prezente pe Web. Pentru unele din ele, rețeaua Web este o afacere profitabilă, datorită vânzărilor online pe care le generează. De exemplu, firme

precum florăriile și bibliotecile online merg bine pe Web. Pentru alte firme, vânzările nu s-au ridicat la nivelul așteptărilor. Multe companii folosesc sistemul Web pentru distribuția informațiilor tehnice (precum remedii ale unor erori, patch-uri și documentație către clienții lor). Părerea generală este că rețeaua Web este un instrument adecvat pentru marketing și servicii destinate clienților.

Folosirea unui server Web într-un intranet

Firmele folosesc acum serverele Web în intraneturi pentru a distribui informații angajaților. Termenul *intranet* se referă la astfel de rețele interne, care folosesc suita de protocoale TCP/IP și furnizează servicii informaționale precum Web și FTP.

Informația publicată într-o rețea internă obișnuită conține informații despre angajați, proceduri și metode comune, evenimente viitoare, locuri de muncă vacante și multe altele. În rețeaua dumneavoastră internă de afaceri puteți partaja documente – documente tehnice, rapoarte de stare, notițe de la ședințe și orice altceva – prin intermediul unui server Web, presupunând că aveți o rețea TCP/IP și browsere Web instalate pe toate sistemele din rețea.

Puteți să setați convenabil un server Web pe un PC Linux pentru a servi ca server al grupului dumneavoastră de lucru și să plasați documentele pe acesta. Va trebui să converțiți documentele formate în formatul HTML; multe programe populare de procesare de texte, inclusiv WordPerfect și Microsoft Word, vă permit să salvați documente în formatul HTML. Apoi, utilizatorii vor fi capabili să acceseze documentele de pe platforma lor de lucru prin intermediul unui browser Web.

În continuare, sunt câteva idei pentru folosirea unui server Web în cadrul unei rețele locale private:

- Furnizează o carte de telefon online, în care angajații pot căuta cu ajutorul unui formular dintr-o pagină Web.
- Permite angajaților să prezinte rapoarte de timp și prezență, cereri și rapoarte cu cheltuielile pentru deplasări, prin intermediul serverului Web.
- Furnizează acces la orice bază de date de firmă, prin intermediul formularelor dintr-o pagină Web.
- Distribuie anunțuri despre ședințe și alte evenimente.
- Implementează un avizier electronic de sugestii, permițând angajaților să-și facă publice sugestiile prin intermediul unui formular dintr-o pagină Web.
- Întreține o listă cu acțiuni pentru diferite proiecte. Furnizează un formular prin care fiecare poate urmări distribuția sarcinilor de serviciu.
- Întocmește documente tehnice importante, disponibile prin intermediul serverului Web.

Linux utilizează eficient resursele hardware, astfel încât puteți implementa un server Web bazat pe Linux pe o arhitectură hardware foarte accesibilă ca preț.

Furnizarea serviciilor Web

Dacă sunteți un furnizor de servicii Internet, puteți folosi PC-uri Linux pentru a oferi servicii Web clienților. Iată câteva servicii Web posibile:

- **Server Web la sediul clientului.** Furnizează clientului un PC Linux și o conexiune dial-up PPP. Configurați PC-ul Linux ca server Web. De asemenea, pregătiți-vă să realizați paginile Web pentru client. În acest caz, lărgimea de bandă a conexiunii dial-up cu sediul clientului

poate fi un factor restrictiv, ceea ce limitează numărul și frecvența accesărilor pe care le poate suporta serverul Web.

■ **Server Web dedicat aflat la sediul ISP-ului.** Oferă clientului un PC Linux dedicat, cu o conexiune de mare viteză la Internet, localizat la sediul dumneavoastră (Presupun că dumneavoastră, ca furnizor de servicii Internet, aveți o conexiune la Internet cu o lărgime de bandă ridicată). Acest scenariu ar trebui să ofere performanțe mult mai bune, deoarece conexiunea la rețea poate susține simultan mai multe accesări Web. Dezavantajul este că trebuie să taxați clientul pentru un PC Linux complet; de asemenea, trebuie să percepeți o taxă pentru accesul la Internet cu o lărgime de bandă ridicată, deoarece oferiți clientului un sistem dedicat. Cu toate că prețul este ridicat, acest serviciu poate fi atractiv pentru companiile mari, care doresc o prezență pe Web fără grija întreținerii unei conexiuni Internet.

■ **Pagini Web individuale.** Furnizează pagini Web individuale pe un PC Linux partajat care activează serverul Web și care are o conexiune la Internet cu o lărgime de bandă înaltă. În acest caz, puteți diviza costul PC-ului Linux și al conexiunii la Internet la mai mulți clienți, astfel încât acest serviciu este recomandat pentru persoane particulare sau mici firme care doresc să fie prezente pe Internet fără să cheltuiască prea mulți bani.

Server LAN Manager

Dacă afacerea dumneavoastră se bazează pe LAN Manager pentru partajarea fișierelor și a imprimantei, probabil că folosiți Windows for Workgroups, Windows NT sau OS/2 pe serverele și clienții dumneavoastră. Puteți folosi un PC Linux ca server fără să pierdeți partajarea fișierelor și a imprimantei, deoarece un PC Linux poate fi setat ca server LAN Manager. Când instalați Linux Red Hat de pe CD-ROM-ul anexat acestei cărți, aveți șansa de a instala pachetul software Samba, care îndeplinește această sarcină.



Sugestie

După ce instalați și configurați Samba pe PC-ul dumneavoastră Linux, PC-urile client (pe care rulează Windows for Workgroups, Windows 95, Windows NT sau OS/2) pot accesa discurile și imprimantele de pe PC-ul Linux folosind protocolul Server Message Block (SMB), care este esențial în administrarea unei rețele locale.

Având instalat pachetul Samba puteți, de asemenea, să faceți din PC-ul dumneavoastră Linux un client LAN Manager, ceea ce înseamnă că PC-ul Linux poate accesa discuri și imprimante gestionate de un server administrator de rețea. De exemplu, imprimanta mea este conectată fizic la un PC pe care rulează Windows 95 și alte PC-uri folosesc această imprimantă prin intermediul partajării imprimantei, asigurată de LAN Manager. Eu folosesc Linux pe un PC care nu are imprimantă proprie. Pentru a tipări de pe PC-ul Linux, folosesc clientul Samba pentru accesarea imprimantei de pe PC-ul pe care rulează Windows 95.

Pachetul software Samba are următoarele componente importante:

- **smbd:** serverul SMB, care acceptă conexiunile cu clienții LAN Manager și furnizează servicii de partajare a fișierelor și imprimantei
- **nmbd:** serverul de nume NetBIOS, pe care clienții îl folosesc pentru a căuta servere. (NetBIOS vine de la Network Basic Input/Output System – o interfață pe care aplicațiile o folosesc să comunice cu protocoalele de transport în rețea precum TCP/IP.)
- **smbclient:** clientul LAN Manager, care rulează pe Linux și permite acestuia să acceseze fișierele și imprimantele de pe orice server LAN Manager
- **/etc/smb.conf:** fișierul de configurare Samba folosit de serverul SMB
- **testparm:** un program care se asigură că fișierul de configurare Samba este corect

Pentru că nu am descris Samba încă, următoarele secțiuni vor descrie modul de instalare de pe CD-ROM-ul anexat acestei cărți și cum să setați o imprimantă de pe PC-ul Linux să tipărească prin intermediul LAN Manager.

Instalarea programului Samba

Dacă ați optat pentru Samba în timpul instalării Linux Red Hat, ar trebui să săriți peste această secțiune. În caz contrar, parcurgeți următorii pași pentru a instala Samba de pe CD-ROM-ul anexat acestei cărți:

1. Intrați în sistem ca *root* și asigurați-vă că CD-ROM-ul este introdus în unitate. Dacă nu este, folosiți comanda `umount /dev/cdrom` pentru a extrage CD-ROM-ul anexat acestei cărți, înlocuiți-l cu cel din această carte și tastați comanda: `mount -r /dev/cdrom /cdrom`.

2. Intrați în directorul de pe CD-ROM – adică în directorul unde sunt localizate pachetele Red Hat Package Manager (RPM):

```
cd /cdrom/RedHat/RPMS
```

3. Folosiți următoarea comandă rpm:

```
rpm -ivh samba*
```

Dacă Samba este deja instalat, această comandă emite un mesaj de eroare.

Altfel, comanda rpm instalează Samba pe sistemul dumneavoastră copiind diferite fișiere în locațiile lor corespunzătoare.

Acești pași încheie decompimarea și instalarea softului Samba. Acum, pentru a putea folosi Samba, trebuie să-l configurați.

Configurarea Samba

Pentru a seta serviciile LAN Manager de partajare ale fișierelor și imprimantei, trebuie să creați un fișier de configurare numit `/etc/smb.conf`. Acesta arată că un fișier cu extensia `.INI` din Microsoft Windows 3.1, în eventualitatea în care sunteți familiarizat cu acele fișiere. Pentru a vă împropăta memoria, următorul exemplu prezintă o parte a fișierului `WIN.INI` din Windows 3.1.

```
[windows]
; Aceasta este secțiunea "windows"
; Liniiile de comentariu încep cu punct și virgula
NullPort=None
load=
run=
device=HP LaserJet 4/4M PostScript,PSCRIPT,LPT1:
```

```
[Desktop]
; Aceasta este secțiunea "Desktop"
Wallpaper=(None)
TileWallpaper=0
WallpaperStyle=0
Pattern=(None)
Multe linii sterse...
```



Secret

Ca și fișierele cu extensia `.INI` din Windows, fișierul `/etc/smb.conf` este format din secțiuni, fiecare conținând o listă de parametri. Fiecare secțiune a fișierului `smb.conf` începe cu numele acesteia inclus între paranteze pătrate. Secțiunea continuă până începe următoarea sau până când se termină fișierul.

Fiecare linie dintr-o secțiune specifică valoarea unui parametru, cu următoarea sintaxă:

```
name = value
```

Ca și în fișierele din Windows cu extensia .INI, liniile de comentariu încep cu semnul punct și virgulă (;). Iată, în continuare, câteva secțiuni tipice din fișierul de configurare Samba:

```
[homes]
; Această secțiune partajează directorul sursa al fiecărui utilizator
comment = Home Directories
browseable = no
read only = no
preserve case = yes
short preserve case = yes
create mode = 0750

[printers]
; Această secțiune specifică partajarea imprimantelor
comment = All Printers
path = /var/spool/samba
browseable = no
printable = yes
; Set public = yes pentru a permite utilizatorului „cont musafir” să tipareasca
public = no
writable = no
create mode = 0700
```

Linii sterse...

Observați asemănările cu fișierele INI din Windows.

Programul Samba este livrat cu un fișier de configurare pe care îl puteți edita pentru a vă începe treaba. Pentru a pregăti fișierul de configurare, parcurgeți următoarele etape:

1. Intrați în sistem ca root.
2. Editați fișierul /etc/smb.conf cu editorul dumneavoastră favorit.
3. În secțiunea global, modificați linia workgroup astfel:
; setați variabila pe numele grupului dumneavoastră de lucru
workgroup = LNB SOFTWARE
4. Ștergeți semnul de comentariu din liniile uneia din secțiunile *public* pentru a furniza acces la un director partajat de pe PC-ul Linux, astfel:

```
[public]
path = /home/public
public = yes
only guest = yes
writable = yes
printable = no
```

Testarea fișierului de configurare Samba

Pentru a fi siguri că fișierul de configurare Samba este corect, lansați în execuție programul testparm care este livrat o dată cu softul Samba. Iată rezultatul rulării programului testparm pentru fișierul meu de configurare:

```
testparm
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Press enter to see a dump of your service definitions (Apasati Enter)
(O lista lunga de parametri si servicii steersa...)
```

Dacă programul testparm raportează erori de sintaxă în fișierul smb.conf, ar trebui să editați fișierul pentru a rezolva problema. Pentru informații detaliate despre conținutul fișierului smb.conf, consultați asistența online tastând comanda `man smb.conf`.



Sugestie

Altă sursă bună de informații despre configurarea aplicației Samba este fișierul SMB-HOWTO. Pentru a citi acest fișier HOWTO, intrați în directorul /usr/doc/HOWTO și tastați `zcat SMB-HOWTO | more` pentru a-l vizualiza.

Testarea cu programul smbclient

Ar trebui să folosiți programul smbclient pentru a vă asigura că serverul LAN Manager funcționează. O metodă rapidă pentru această verificare este folosirea opțiunii `-L` pentru vizualizarea listei de servicii. Iată mesajul emis atunci când lansez în execuție smbclient pe PC-ul meu Linux:

```
smbclient -L dcc05211
Server time is Fri May 22 07:11:40 1998
Timezone is UTC-4.0
Password:
Domain=[LNB SOFTWARE] OS=[Unix] Server=[Samba 1.9.17p4]
security=user
```

```
Server=[DCC05211] User=[naba] Workgroup=[LNB SOFTWARE] Domain=[LNB SOFTWARE]
```

Sharename	Type	Comment
IPC\$	IPC	IPC Service (Samba 1.9.17p4)
Lp	Printer	
Naba	Disk	Home Directories
Sample	Printer	A sample printer that prints to a file

This machine has a browse list:

Server	Comment
DCC05211	Samba 1.9.17p4

This machine has a workgroup list:

Workgroup	Master
LNB SOFTWARE	DCC05211

Numele serverului vine de la numele asociat cu adresa IP a interfeței Ethernet a PC-ului meu Linux. Acest nume apare în fișierul /etc/hosts.

Dacă aveți alte servere LAN Manager prin apropiere, vă puteți uita la serviciile lor cu programul smbclient. Iată mesajul pe care-l primesc când vizualizez serviciile de pe PC-ul meu 486 pe care rulează Windows 95:

```
smbclient L lnb486
Server time is Thu May 21 19:34:52 1998
Timezone is UTC-4.0
security=share
Server=[LNB486] User=[ ] Workgroup=[LNB SOFTWARE] Doma în=[LNB SOFTWARE]
```

Sharename	Type	Comment
A	Disk	

```

C          Disk
D          Disk
E          Disk
F          Disk
HPLJ4M     Printer    Hp LaserJet 4M on Dell 486
IPC$       IPC        Remote Inter Process Communication
PRINTERS$  Disk

```

This machine has a workgroup list:

```

Workgroup      Master
-----
LNB SOFTWARE   LNB486

```



Secret

Puteți face mult mai multe cu programul `smbclient` decât simpla examinare a resurselor; îl puteți folosi, de asemenea, pentru a accesa un disc de pe un server LAN Manager, precum și pentru a trimite un fișier către o imprimantă LAN Manager. Programul `smbclient` este asemănător cu `ftp` – vă conectați la un server LAN Manager și apoi folosiți comenzi pentru a încărca sau descărca fișiere pe acesta și pentru a trimite fișiere către imprimantă.

Următorul exemplu arată cum am folosit programul `smbclient` pentru a accesa un disc de pe PC-ul meu Windows 95 și pentru a-i vedea directoarele:

```

smbclient \ \ \ \ lnb486 \ \ f username password
Server time is Thu May 21 21:04:50 1998
Timezone is UTC-4.0
security=share
smb: \> dir
-MSSTFQF.T          DHR      0 Mon Jan 16 10:12:42 1995
WINDOWS.000         D          0 Fri Nov 25 18:09:14 1994
MSVCRT20.DLL        A    243200 Fri Oct 28 12:00:00 1994
RECYCLED            DHS      0 Sat Nov 26 22:51:00 1994
Autoexec.bat        A       54 Sun Mar 9 12:16:22 1997
Ffastun.ffl         AH    32768 Thu Feb 12 20:00:42 1998
Program Files       DR      0 Sat Mar 18 16:44:38 1995
Ffastun0.ffx        AH   114688 Thu Feb 12 20:00:42 1998
Ffastun.ffa         AH    24576 Thu Feb 12 20:00:42 1998
Ffastun.ffa         AH     4379 Thu Feb 12 20:00:42 1998

```

51141 blocks of size 4096. 3440 blocks available

smb: \> quit

Pentru a vedea o listă a comenzilor `smbclient`, tastați `help` la prompt-ul afișat. Tabelul 21-1 prezintă pe scurt comenzile `smbclient`.

Tabelul 21-1 Câteva comenzi `smbclient`

Comandă	Descriere
<code>!</code>	Execută o comandă de interpretor (nu uitați că rulați <code>smbclient</code> pe Linux)
<code>? cmd</code>	Afișează o listă de comenzi sau asistență despre o anumită comandă
<code>cancel id</code>	Anulează o acțiune de tipărire desemnată prin identificatorul ei
<code>cd dir</code>	Schimbă directorul distant
<code>del file</code>	Șterge fișierul specificat
<code>dir file</code>	Afișează lista directoarelor

Comandă	Descriere
<code>exit</code>	Deconectează serverul LAN Manager
<code>get rfile lfile</code>	Copiază un fișier distant (<i>rfile</i>) într-un fișier local (<i>lfile</i>)
<code>help cmd</code>	Furnizează asistență despre o comandă (sau afișează o listă de comenzi)
<code>lcd newdir</code>	Schimbă directorul curent (de pe PC-ul Linux)
<code>lowercase</code>	Activează automat convertirea numelor de fișiere în litere mici când se execută comanda <i>get</i>
<code>ls files</code>	Listează fișierele de pe server
<code>mask name</code>	Aplică o mască (precum <i>*.c</i>) tuturor operațiilor cu fișiere
<code>md dirname</code>	Creează un director pe server
<code>mget name</code>	Selectează toate fișierele ce au o porțiune de nume identică (ca de exemplu <i>*.doc</i>)
<code>mkdir dirname</code>	Creează un director
<code>mput name</code>	Copiază fișiere de pe PC-ul Linux pe server
<code>newer file</code>	Selectează numai acele fișiere mai noi decât fișierul specificat
<code>print name</code>	Tipărește fișierul specificat
<code>printmode mode</code>	Setează modul de tipărire (<i>mode</i> trebuie să fie <i>text</i> sau <i>graphics</i>)
<code>prompt</code>	Comută promptul mode pe off (asemănătoare cu comanda <i>ftp</i>).
<code>put lfile rfile</code>	Copiază un fișier local (<i>lfile</i>) într-un fișier distant (<i>rfile</i>)
<code>queue</code>	Afișează coada de tipărire
<code>quit</code>	Deconectează serverul LAN Manager
<code>rd dir</code>	Șterge directorul specificat de pe server
<code>recurse</code>	Comută recursivitatea directoarelor în timpul operațiilor de încărcare și descărcare a fișierelor
<code>rm name</code>	Șterge toate fișierele cu numele specificat
<code>rmdir name</code>	Șterge directorul specificat
<code>translate</code>	Comută pe traducerea textului (convertește saltul la linie nouă la o pereche retur de car – salt la linie nouă)

Client LAN Manager

Dacă aduceți un PC Linux într-un mediu LAN Manager existent, va trebui să-l folosiți drept client LAN Manager, deoarece poate exista un server. Așa cum explică secțiunile precedente, pachetul de software Samba conține programul `smbclient`, care permite PC-ului dumneavoastră Linux să fie client în rețeaua LAN Manager.



Referință încrucișată

În această secțiune, este prezentat modul de setare a unei imprimante pe PC-ul dumneavoastră Linux astfel încât sarcinile de tipărire sunt trimise la un anumit server LAN Manager prin intermediul programului `smbclient`. Capitolul 14 descrie cum funcționează sistemul de tipărire în Linux. Dacă aveți întrebări despre tipărire în timp ce citiți această secțiune, vă rog să consultați capitolul respectiv.

Setarea imprimantei în două etape

Ideea de bază este de a defini o nouă imprimantă pe PC-ul dumneavoastră Linux astfel încât atunci când tipăriți folosind comanda `lpr`, rezultatul apare de fapt pe o anumită imprimantă din rețea (gestionată de serverul LAN Manager). Acest proces necesită două etape:

1. Adăugați o intrare pentru imprimantă în fișierul `/etc/printcap` și specificați un filtru de intrare (un script care procesează fișierul ce este tipărit).
2. Scrieți fișierul script care lansează în execuție programul `smbclient` și tipărește fișierul pe un server LAN Manager dedicat.

Adăugarea intrării printcap

Când pregătiți intrarea `printcap` care tipărește pe o imprimantă LAN Manager, observați două puncte cheie:

- Specificați `/dev/null` ca nume al imprimantei, deoarece tipărirea se face pe o imprimantă din rețea.
- Specificați un filtru care copiază datele de intrare într-un fișier și apoi folosește `smbclient` pentru a trimite acel fișier către imprimanta conectată la un server LAN Manager selectat.



Iată intrarea `printcap` (din fișierul `/etc/printcap`) pe care am folosit-o pe PC-ul meu Linux pentru a specifica această imprimantă:

Secret

```
lp|smbpr|Prints on a LAN Manager printer:\
:sd=/var/spool/lpd/lp:\
:lp=/dev/null:\
:if=/var/spool/lpd/lp/smbprint.tcl:\
:if=/var/spool/lpd/lp/smbprint.log:\
:mx#0:\
:sh:\
:Sf:
```

Pentru erori de conectare, am creat fișierul `/var/spool/lpd/laptop/smbprint.log` cu următoarele comenzi (în timp ce eram conectat ca `root`):

```
c d /var/spool/lpd/lp
touch smbprint.log
chmod ug=rw, o=r smbprint.log
```

Scrierea unui fișier script pentru a tipări cu programul smbclient

Următorul pas important este scrierea filtrului de intrare – `/var/spool/lpd/laptop/smbprint.tcl`, care realizează efectiv tipărirea. După cum sugerează și extensia, am folosit limbajul script Tcl pentru a scrie filtrul de intrare. Puteți scrie fișierul script în limbajul Perl sau bash.

În fișierul script `smbprint.tcl`, am hotărât să fac o copie a lucrării de tipărire într-un fișier temporar și apoi să lansez în execuție `smbclient` pentru a tipări fișierul. În continuare este prezentat fișierul script complet:

```
#!/usr/bin/tcl
#
# Script Tcl pentru tipărirea pe o imprimanta LAN manager folosind smbclient.
# Plasati-l în directorul spool al imprimantei si conectati-va la Internet.
```

```
# Deschideti fisierul /tmp/smbprint.out
set outfile [open /tmp/smbprint.out w]

# Cititi din stdin si scrieti în outfile

While { [gets stdin line] != -1 } {
    puts $outfile $line
}

close $outfile

# Acum pregatiti un script pentru a lansa în executie programul smbclient și
tipariti fisierul
set server "lmb486"
set printer "hplj4m"
set password "mypassword"

set outfile [open /tmp/printit w]
puts $outfile "# !!/bin/sh"
putd $outfile "echo print /tmp/smbprint.out | /usr/bin/smbclient
\\\\\\\\\\\\\\\\$serve
r \\\\\\\\\$printer $password -N -P"
close $outfile

# Faceti fisierul script executabil
exec chmod +x /tmp/printit

# Acum executati scriptul
exec /tmp/printit
# Stergeti scriptul
exec rm /tmp/printit
```

Am scris fișierul script într-un mod simplu: am copiat lucrarea de tipărire într-un fișier temporar, am creat un script ce lansează în execuție `smbclient` pentru a tipări acel fișier temporar și apoi am rulat acest script. Am folosit această setare a imprimantei pentru a tipări direct din programele Linux (precum Netscape) pe acea imprimantă din rețea.



Observație

Acest capitol oferă o imagine de ansamblu a programului Samba, dar nu dispunem de spațiu suficient pentru a vă oferi toate informațiile disponibile. Pentru informații suplimentare despre modul de utilizare a softului Samba consultați cartea *Samba: Integrating UNIX and Windows*, SSC, 1998, scrisă de John Blair.

Linux în anumite domenii

În secțiunile anterioare, am descris câteva activități specifice pentru care Linux este soluția potrivită. Următoarele secțiuni se concentrează asupra afacerilor specifice pentru care Linux este instrumentul de bază. Când descriu rolul sistemului Linux într-o afacere, menționez, de asemenea, operațiile posibile cu Linux.

Furnizor de servicii Internet

Furnizorul de servicii Internet (ISP) este un tip de afacere relativ nouă, care a apărut datorită faptului că tot mai multe persoane fizice și companii vor să se conecteze la Internet. Multe ISP-uri au început prin a oferi conturi dial-up prin modem utilizatorilor individuali la începutul anilor '90. Apoi, pe măsură ce Internetul a devenit mai popular, ISP-urile au început să ofere conexiuni SLIP și PPP în ISDN (Integrated Services Digital Network, o ofertă a companiilor telefonice) și alte conexiuni de mare viteză ce oferă acces total în rețele TCP/IP. (Consultați capitolul 18 pentru mai multe informații despre SLIP și PPP.)

În prezent, cele mai multe ISP-uri întrețin una sau mai multe linii T1 cu marile companii de telecomunicații, precum Sprint și MCI. (O legătură T1 este capabilă de a transmite 1,54 milioane de biți de date pe secundă – de peste 50 de ori mai rapidă decât rata de transfer de 28800 bps a modemurilor). Clienții individuali au, de obicei, conexiuni SLIP/PPP prin modemuri dial-up de mare viteză.

Pe lângă diferitele echipamente de telecomunicații (inclusiv modemurile), ISP-urile au nevoie de calculatoare gazdă în Internet, în speță servere care manevrează conexiunile SLIP/PPP precum și conexiunile simple dial-in. Aici intervine Linux. Cu facilitățile interne de lucru în rețele TCP/IP și suportul pentru toate tipurile de componente hardware de rețea, PC-urile Linux sunt ideale pentru acest aspect foarte important al activității unui ISP. Multe ISP-uri consideră că PC-urile Linux sunt calculatoare gazdă în Internet foarte performante.



Observație

Am folosit Windows Netscape Navigator, SunOS, HP-UX și Linux și consider că acesta din urmă este cel mai flexibil și mai complet sistem de operare. Nu pot spune că celelalte sisteme de operare nu sunt bune, dar aproape totul – de la un compilator de C sau C++ până la suport PPP – este, de obicei, o opțiune. Așa cum probabil v-ați dat seama, fiecare opțiune costă bani. Dacă vreți să deveniți ISP (furnizor de servicii Internet), bazați-vă pe PC-urile Linux.



Sugestie

Cu toate că descriu în această secțiune câteva aspecte ale afacerii unui ISP, ar trebui să navigați pe Internet și să citiți informațiile disponibile înainte de a vă decide să deveniți ISP. Un punct de plecare recomandat este Internet Access Frequently Asked Questions, întreținut de David Dennis, la adresa

<http://www.amazing.com/internet/>.

În respectiva pagină Web veți găsi informații detaliate despre toate aspectele unei afaceri ISP, de la tipuri viabile de conexiuni de rețea până la întocmirea facturilor și contabilitate.

Ca ISP nu vă puteți descurca numai cu PC-uri Linux. Iată care sunt echipamentele de care aveți nevoie pentru a porni o afacere ca ISP:

- *O conexiune la Internet cu lărgime mare de bandă.* Ca ISP, veți apela pentru acest serviciu la una din marile companii de telecomunicații. Pentru o scurtă perioadă de început, această conexiune poate fi de 56 Kbps, dar o conexiune T1 de 1,54 Mbps este mai adecvată pentru un adevărat ISP.
- *Un router* pentru a direcționa pachetele de rețea între sistemul dumneavoastră și restul Internetului. Cu toate că Linux poate lucra ca router, sarcina este îndeplinită cel mai bine de un router dedicat.
- *Echipament de rețea* (precum CSU/DSU) pentru a vă conecta la compania de telecomunicații.
- *Un server terminal* care să se ocupe de modemurile dial-up precum și de conexiunile SLIP și PPP. Un PC Linux cu placă serială cu porturi multiple poate îndeplini această sarcină, dar conform filosofiei „instrumentul potrivit pentru activitatea potrivită” trebuie să folosiți un server terminal pentru această sarcină. Manevrarea modemurilor dial-up multiple necesită atenția constantă a procesorului. Dacă PC-ul Linux este ocupat cu datele sosite de la modemuri, nu poate să îndeplinească în mod adecvat alte sarcini. Pe de altă parte, un server terminal cu un procesor dedicat poate manevra cu ușurință modemurile dial-up.
- *Linii telefonice pentru acces dial-up*, care reprezintă modul cel mai frecvent prin care clienții dumneavoastră vă accesează sistemul. Pentru un mic grup de utilizatori, vă trebuie o linie telefonică pentru fiecare 6-8 utilizatori; pentru mai mulți utilizatori, se recomandă o linie pentru fiecare 10 utilizatori.

- *PC-uri server* (calculatoare gazdă în Internet) pentru mesaje, poștă, World Wide Web și conectările utilizatorilor. Aici se folosesc PC-urile Linux.
- *Surse neîntreruptibile de tensiune* (UPS) pentru a continua lucrul sau pentru a opri sistemul în mod corespunzător în timpul căderilor de tensiune.

Așa cum o arată această listă de echipamente, PC-urile Linux sunt cele mai folosite servere. De obicei, aveți nevoie doar de câteva servere, fiecare dintre ele fiind dedicat unui serviciu specific precum World Wide Web și grupuri de discuții Usenet.

Dezvoltator de programe în UNIX

Dacă afacerea dumneavoastră dezvoltă aplicații pentru UNIX și X Window System ar trebui să aveți propriul dumneavoastră know-how tehnic pentru a instala și seta Linux pe PC-uri. Deoarece Linux este livrat cu un set complet de instrumente pentru dezvoltarea aplicațiilor de software, un PC Linux poate servi ca stație de lucru pentru orice programator.

Cu toate că programul Linux este livrat cu multe limbaje de programare, un PC Linux este ideal pentru programarea aplicațiilor Linux. În prezent, cele mai multe aplicații sunt grafice și în Linux puteți folosi XFree86 X Window System pentru grafică. Programul Linux de pe CD-ROM-ul anexat acestei cărți toate fișierele antet necesare și bibliotecile pentru programarea aplicațiilor X.



Sugestie

Ceea ce nu veți găsi pe CD-ROM este Motif – interfața grafică utilizator bazată pe X. Din păcate, Motif este un produs cu licență Open Software Foundation și va trebui să-l cumpărați separat. Mai multe firme distribuie Motif pentru Linux; prețurile sunt cuprinse între 100 și 200 de dolari. (Vezi capitolul 23 pentru informații suplimentare.)

PC-urile Linux pot fi stații de lucru UNIX cu preț redus, ideale pentru studenți. În loc să partajeze un sistem VAX sau alt sistem de operare mare cu alți utilizatori, fiecare student poate avea o stație de lucru completă cu PC-uri Linux.

Consultant

Cum am explicat anterior în acest capitol, în multe domenii nu se folosește Linux, deoarece nu există nici o sursă de suport tehnic. Puteți deveni consultant Linux pentru a vinde afaceri bazate în principal pe Linux și apoi să furnizați suportul tehnic necesar.

Iată câteva modalități prin care puteți oferi servicii înrudite cu Linux pentru anumite afaceri:

- Organizați scurte cursuri despre instalarea și setarea Linux, precum și setarea unui calculator gazdă Linux în Internet și a unui server Web
- Livrați PC-urile Linux cu suport tehnic consistent ca soluții pentru nevoile specifice ale clienților, cum ar fi conectarea la Internet sau prezența în paginile Web.



Sugestie

În loc de a oferi numai suport tehnic pentru Linux, puteți considera crearea unui pachet integrat de soluții bazat pe Linux. Pentru o anumită necesitate a unei afaceri, ar trebui să fiți capabil de a configura PC-ul Linux cu programele gratuite adecvate din Internet. Apoi, puteți vinde această afacere ca un pachet ce include hardware, software și suport tehnic.

Dacă lucrați cu afaceri care se bazează pe Windows for Workgroups, propuneți un PC Linux cu programul Samba ca înlocuitor pentru serverul LAN Manager. Din nou, vindeți serverul Linux LAN Manager ca un sistem la cheie, cu suport tehnic.

Rezumat

Lumea afacerilor ezită deseori să folosească Linux, pentru că nu este un produs cu suport tehnic. Linux are, de asemenea, reputația de a fi ideal pentru hackeri și nu dispune de nici o aplicație personală foarte cunoscută, folosită curent (precum procesoarele de texte și foile de calcul tabelar). În ciuda acestor neajunsuri, Linux are foarte multe de oferit, atâta timp cât este folosit în scopuri adecvate. Acest capitol descrie rolul și câteva utilizări în afaceri ale sistemului Linux.

Din lectura acestui capitol, ați învățat următoarele:

- ▶ Cu toate că inițial sistemului Linux i-au lipsit aplicațiile de birou, precum procesoarele de texte și foile de calcul tabelar, acum situația s-a schimbat. Puteți alege dintre câteva pachete de aplicații de birou incluzând StarOffice, ApplixWare și WordPerfect pentru Linux.
- ▶ Linux excelează în rețele și instrumente pentru dezvoltare de software. Pentru a înlătura orice îndoială asupra posibilităților lui, Linux poate rula pe PC-uri obișnuite, care sunt mult mai ieftine decât stațiile de lucru Sun și HP. Astfel, PC-urile Linux sunt ideale ca sisteme gazdă în Internet și ca stații de lucru pentru dezvoltare de aplicații software.
- ▶ Pachetul Linux Red Hat de pe CD-ROM-ul anexat acestei cărți include aproape toate instrumentele pentru lucrul în rețea și pentru dezvoltarea aplicațiilor software necesare unei afaceri, pentru un preț nominal (în acest caz, prețul cărții de față).
- ▶ Pentru rețelele de PC-uri ce folosesc protocolul LAN Manager, puteți configura un PC Linux ca server LAN Manager; trebuie doar să folosiți pachetul software Samba de pe CD-ROM-ul anexat. Acest capitol vă arată cum să instalați și să folosiți Samba.
- ▶ Un PC Linux se poate, de asemenea, comporta ca un client LAN Manager. Ca exemplu, acest capitol descrie utilizarea unei imprimante LAN Manager pe un PC Linux.
- ▶ PC-urile Linux sunt deosebit de potrivite pentru a fi servere pentru furnizorii de servicii Internet. Facilitatea TCP/IP încorporată în Linux este un mare avantaj pentru un ISP.
- ▶ Lipsa binecunoscută a suportului Linux poate fi o oportunitate de afaceri pentru cei care pot instala și întreține Linux. Altă afacere potențială este de a vinde PC-uri Linux configurate cu programe gratuite sau la cererea clienților și destinate îndeplinirii unei anumite necesități. Dacă sunt dotate și cu suport tehnic, astfel de PC-uri Linux pot fi considerate soluții de afaceri nu doar sisteme UNIX a căror singură calitate este prețul redus.

Capitolul 22

Programarea în Linux

În acest capitol

- ▶ Veți examina GCC: compilatoarele GNU C și C++
- ▶ Veți folosi utilitarul GNU make pentru a automatiza generarea aplicațiilor software
- ▶ Veți folosi depanatorul GNU: gdb
- ▶ Veți folosi RCS pentru controlul versiunii
- ▶ Veți înțelege licențele GPL și LGPL
- ▶ Veți înțelege formatul ELF
- ▶ Veți încărca și veți folosi o bibliotecă partajată în mod dinamic

Se întâmplă ca mulți utilizatori Linux să fie dezvoltatori de software. Dacă doriți să dezvoltați software ca un hobby sau doriți să adăugați noi facilități Linux, veți observa că acesta conține toate cele necesare pentru a crea aplicații UNIX și X. Puteți folosi compilatoarele GNU C și C++ pentru a scrie programe convenționale (pe care le compilați și le transformați în executabile). Ca o metodă alternativă, puteți folosi limbajul script Tcl/Tk pentru a scrie aplicații grafice interpretate.

Acest capitol acoperă subiectul general al programării pe un PC Linux. Accentul nu este pus pe un anumit limbaj de programare. În schimb, acest capitol descrie cum să folosiți diferite instrumente de programare, cum ar fi compilatoarele, fișierele `makefile` și sistemele pentru controlul versiunii.

De asemenea, vom discuta despre subiectul legării dinamice și despre recent adoptatul Executable and Linking Format (ELF), care realizează legăturile dinamice mai ușor. Aceste subiecte sunt interesante pentru programatorii în Linux, deoarece legarea dinamică reduce mărimea fișierelor executabile și permite distribuirea propriului dumneavoastră software în format binar, chiar dacă programul dumneavoastră folosește bibliotecile GNU.



Referință
încrucișată

Dezvoltarea aplicațiilor X este un subiect destul de complex pentru a acoperi un întreg capitol. În consecință, programarea aplicațiilor X în Linux este tratată în capitolul 23.

Instrumente pentru dezvoltare software în Linux

Așa cum era de așteptat, fiind asemănător cu UNIX, Linux include instrumentele tradiționale UNIX pentru dezvoltare de software. Prin instrumente tradiționale, mă refer la următoarele:

- Un editor de text cum ar fi `vi` sau `emacs`, pentru editarea codului sursă (descriș în capitolul 24)
- Un compilator C pentru compilarea și legarea programelor scrise în C – limbajul de programare ales pentru scrierea aplicațiilor UNIX (cu toate că, în prezent, mulți programatori folosesc C++). Linux conține GCC: compilatoarele GNU C și C++.

- Facilitatea GNU make pentru automatizarea procesului de generare a softului (*software build*) – procesul combinării modulelor din obiecte într-un fișier executabil sau într-o bibliotecă.
- Un utilitar pentru depanarea programelor. Linux conține depanatorul GNU gdb, precum și *xxgdb*, o interfață grafică pentru gdb.
- Un sistem pentru controlul versiunii, pentru a ține evidența diferitelor revizii aduse unui fișier sursă. Linux este livrat cu RCS (Revision Control System).



CD

Toate aceste instrumente sunt instalate automat dacă instalați Linux de pe CD-ROM-ul anexat acestei cărți, urmând etapele descrise în capitolul 1. Următoarele secțiuni descriu pe scurt cum se folosesc aceste instrumente pentru a scrie aplicații pentru Linux.

info: Documentația pentru asistența de încredere pentru instrumentele GNU

Probabil că ați observat deja că majoritatea instrumentelor de dezvoltare software din Linux își au originea în proiectul GNU al Free Software Foundation. Documentația online pentru toate aceste instrumente este sub formă de fișiere *info*. Programul *info* este sistemul de asistență al hipertextului din GNU.



Sugestie

Pentru a vedea *info* în acțiune, tastați *info* la prompt-ul interpretorului sau tastați *Esc-x* urmat de *info* în GNU Emacs. De obicei, eu accesez *info* din GNU Emacs, deoarece în acest fel pot apela asistența online în timp ce editez un program.

Figura 22-1 prezintă fereastra GNU Emacs după ce am tastat *Esc-x info*.

În *info*, textul asistenței online este organizat în noduri; fiecare nod reprezintă informații despre un anumit subiect. Prima linie conține antetul nodului.

Figura 22-1

Fereastra GNU Emacs, cu rezultatul comenzii *Esc-x info*.

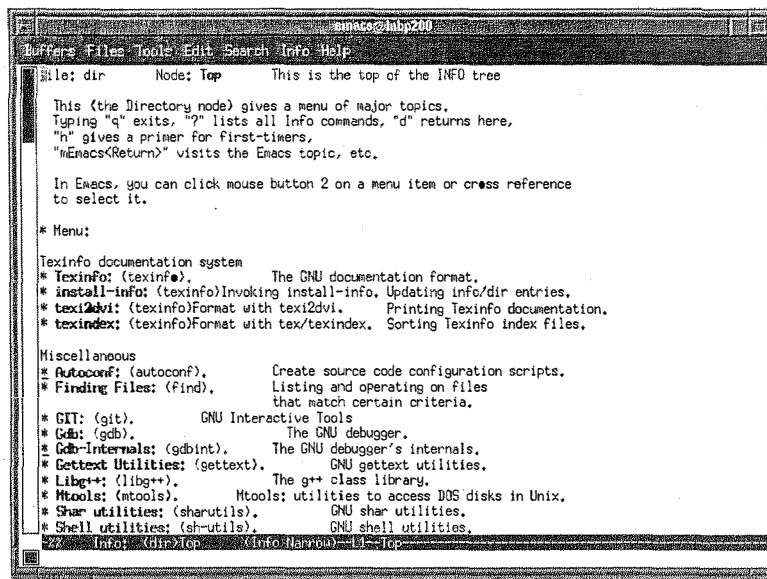


Figura 22-1 conține ecranul inițial *info*, cu un director de subiecte. Acest director este un fișier *info*, */usr/info/dir*, un fișier text conținând câteva caractere speciale înglobate. Iată câteva linii din fișierul */usr/info/dir*, corespunzătoare ecranului din figura 22-1:

```
$Id: dir,v 1.2 1996/09/24 18:43:01 karl Exp $
This is the file.../info/dir, which contains the topmost node of the
Info hierarchy. The first time you invoke Info you start off
looking at that node, which is (dir)Top.
^_ (Acesta este caracterul Ctrl+_)
```

```
File: dir Node: Top This is the top of the INFO tree
```

```
This (the Directory node) gives a menu of major topics.
Typing „q” exits, „?” lists all Info commands, „d” returns here,
„h” gives a primer for first-timers,
„mEmacs<Return>” visits the Emacs topic, etc.
```

In Emacs, you can click mouse button 2 on a menu item or cross reference to select it.

```
* Menu:
* Texinfo documentation system
* Texinfo: (texinfo). The GNU documentation format.
* install-info: (texinfo) Invoking install-info. Updating info/dir
entries.
* texi2dvi: (texinfo) Format with texi2dvi. Printing Texinfo
documentation.
* texindex: (texinfo) Format with tex/texindex. Sorting Texinfo
index files.
(...Linii sterse...)
```

În cazul în care comparați acest listing cu ecranul din figura 22-1, observați că *info* afișează numai liniile care urmează după caracterul *Ctrl+_*. În sistemul dumneavoastră, directorul */usr/info* conține acest fișier *info* precum și altele, cu textul pentru fiecare subiect. Aceste fișiere *info* sunt stocate, de obicei, într-un format comprimat. Nu trebuie neapărat să cunoașteți aceste detalii pentru a folosi fișierele *info*.



Sugestie

Trebuie să folosiți câteva comenzi dintr-o singură literă pentru a vă deplasa printre fișierele *info*. Calea cea mai bună de a învăța comenzile este să tastați *h* din directorul inițial *info*, prezentat în figura 22-1. După citirea ecranelor de asistență, tastați *d* pentru a reveni la directorul inițial de subiecte.

În ecranul prezentat în figura 22-1, tastați *m* urmat de numele unui articol din meniu (scris cu caractere aldine și având ca prefix un asterisc). Pentru a vedea asistența online pentru GCC, de exemplu, tastați *m* și apoi *gcc*. Sistemul *info*, la rândul lui, afișează meniul de la nivel maxim cu articole pentru GCC, ca în figura 22-2.

Puteți continua explorările tastând *m*, urmat de unul din articolele din meniu prezentate în figura 22-2.

În acest punct, puteți apăsa tasta spațiu din ecranul arătat în figura 22-2. Această acțiune afișează GNU General Public License (GPL), prezentată în figura 22-3.

Linux și compilatorul *gcc* intră sub incidența GPL, care necesită distribuția codului sursă (de aceea, toate versiunile Linux sunt livrate cu fișierele sursă). În secțiunea „Implicațiile licențelor GNU”, veți învăța că puteți folosi instrumente GNU pentru a dezvolta aplicații comerciale și puteți să le distribuiți în format binar (fără cod sursă) atât timp cât ele se leagă numai cu bibliotecile GNU selectate.

Figura 22-2

Fereastra GNU Emacs, prezentând textul de asistență de nivel maxim pentru GCC.

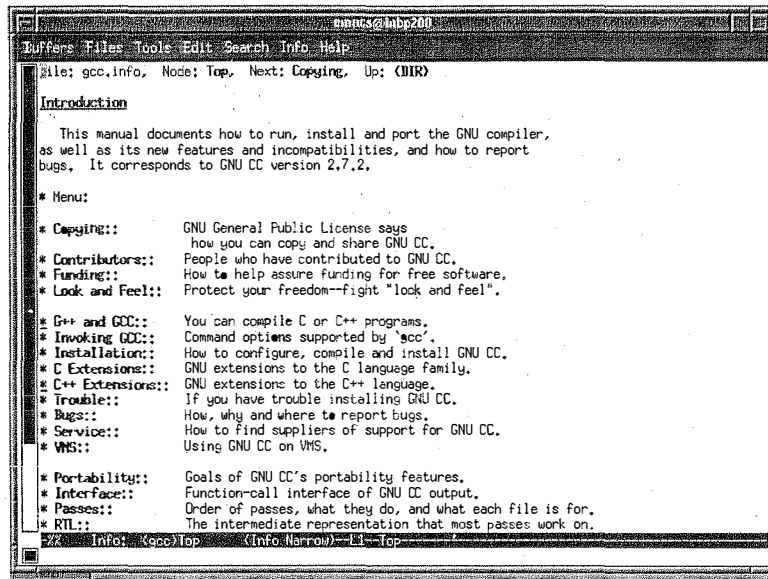
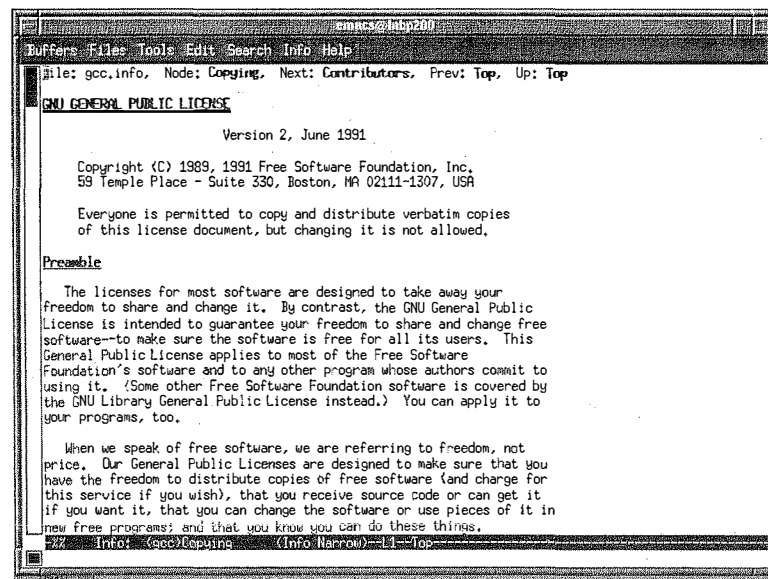


Figura 22-3

Fereastra GNU Emacs, cu prima pagină a GNU General Public License (GPL).



În orice moment, în info puteți tasta **d** pentru a reveni la directorul info cu subiecte, prezentate în figura 22-1. Din acel ecran, puteți examina textul de asistență pentru alte instrumente GNU, precum make și depanatorul GNU.

Pentru a părăsi info, tastați **q**. Dacă ați accesat info din editorul Emacs, tastați **Ctrl+X@** (**Ctrl+C** pentru a părăsi editorul Emacs).

Compilatoarele C și C++ GNU

Cel mai important instrument de dezvoltare software în Linux este GCC, care reprezintă compilatoarele C și C++ GNU. De fapt, GCC poate compila trei limbaje de programare: C, C++ și Objective-C (un limbaj C care are extensii orientate pe obiecte). Pentru a compila și lega fișiere sursă C și C++ folosiți aceeași comandă gcc. Compilatorul GCC suportă standardul ANSI C, astfel încât orice program ANSI C poate fi compilat cu ușurință în Linux. În plus, dacă ați folosit vreodată un compilator C pe alte sisteme UNIX, atunci veți considera GCC foarte simplu.

Apelarea GCC

Folosiți comanda gcc pentru a invoca compilatorul GCC. Prestabilit, când folosiți comanda gcc pentru un fișier sursă, GCC preprocesează, compilează și leagă fișierul executabil. Puteți folosi opțiunile GCC pentru a opri acest proces într-un stadiu intermediar. De exemplu, puteți invoca comanda gcc cu opțiunea **-c** pentru a compila un fișier sursă și a genera un fișier obiect, dar fără să realizați legarea.

Utilizarea GCC pentru compilarea și legarea câtorva fișiere sursă C este ușoară. Să presupunem că doriți să compilați și să legați un program simplu, alcătuit din două fișiere sursă. Următorul listing prezintă fișierul `area.c`, programul principal care calculează aria unui cerc având raza specificată în linia de comandă:

```
#include <stdio.h>
#include <stdlib.h>

/* Functia prototip */
double area_of_circle (double r);

void main(int argc, char **argv)
{
    if (argc < 2)
    {
        printf("Usage: %s radius\n", argv[0]);
        exit(1);
    }
    else
    {
        double radius = atof(argv[1]);
        double area = area_of_circle (radius);
        printf ("Area of circle with radius %f = %f\n",
                radius, area);
    }
}

Listingul următor prezintă fișierul circle.c, care definește o funcție ce calculează aria unui cerc.

#include <math.h>

#define SQUARE(x) ( (x)*(x) )

double area_of_circle (double r);
{
    return 4.0 * SQUARE (r);
}
```

Pentru un program atât de simplu, desigur, aș fi putut să introduc totul într-un fișier, dar acest exemplu ipotetic vă arată cum să manevrați mai multe fișiere.

Pentru a compila aceste două programe și a crea un fișier executabil numit `area`, puteți folosi următoarea comandă:

```
gcc -o area area.c circle.c
```

Această invocare a compilatorului GCC folosește opțiunea `-o` pentru a specifica numele fișierului executabil. (Dacă nu specificați numele unui fișier rezultat, GCC creează un fișier numit `a.out`.)

Dacă aveți multe fișiere sursă de compilat și legat, veți dori să le compilați individual și să generați fișiere obiect (cu extensia `.o`). În acest fel, când modificați un fișier sursă, trebuie să compilați numai acel fișier și apoi să legați toate fișierele obiect. Următorul exemplu arată cum puteți separa etapele de compilare și legare pentru programul de exemplificare:

```
gcc -c area.c
gcc -c circle.c
gcc -o area area.o circle.o
```

Primele două invocări ale `gcc` cu opțiunea `-c` compilează fișierele sursă. A treia invocare leagă fișierele obiect într-un fișier executabil numit `area`.

În caz că sunteți curios, iată cum se rulează programul de exemplificare (pentru a calcula aria unui cerc cu rază unitară):

```
area 1
Area of circle with radius 1.000000 = 12.566371
```

Compilarea programelor C++

GNU CC este un compilator combinat de C și C++, astfel încât comanda `gcc` poate compila și fișiere sursă C++. GCC folosește extensia fișierelor pentru a determina dacă un fișier este C sau C++. Fișierele C au extensia `.c`, în timp ce fișierele C++ se termină cu `.c` sau `.cpp`.

Cu toate că poate compila un fișier C++, comanda `gcc` nu realizează automat legătura cu diferite biblioteci de care programele C++ au nevoie. De aceea este mai ușor de compilat și legat un program C++ folosind comanda `g++`, care invocă `gcc` cu opțiunile adecvate.

Să presupunem că doriți să compilați următorul program C++ simplu, stocat într-un fișier numit `hello.c` (se obișnuiește să se folosească extensia C cu majusculă pentru fișierele sursă C++):

```
#include <iostream.h>

void main (void)
{
    cout << „Hello from Linux!“ << endl;
}
```

Pentru a compila și lega acest program într-un program executabil numit `hello`, folosiți următoarea comandă:

```
g++ -o hello hello.c
```

Această comandă creează executabilul `hello`, pe care-l puteți rula astfel:

```
hello
Hello from Linux!
```

După cum veți vedea în secțiunea următoare, o multitudine de opțiuni GCC controlează diferite aspecte ale compilării programelor C++.

Explorarea opțiunilor compilatorului GCC

Sintaxa de bază a comenzii `gcc` este următoarea:

```
gcc options filenames
```

Fiecare opțiune începe cu o liniuță (`-`) și are de obicei un nume lung, precum `-funsigned-char` sau `-finline-functions`. Multe opțiuni folosite uzual sunt scurte, ca de exemplu `-c` numai pentru compilare și `-g` pentru a genera informații de depanare (necesare pentru depanarea programului cu depanatorul GNU).



Sugestie

Puteți vedea un rezumat al tuturor opțiunilor GCC folosind comanda `info`. Tastați `info` la prompt-ul interpretorului și tastați `m` urmat de `gcc`. Apoi urmăriți articolele din meniu: Invoking GCC -> Option Summary. De obicei, nu trebuie să specificați opțiunile GCC explicit; setările prestabilite sunt adecvate pentru majoritatea aplicațiilor. Tabelul 22-1 listează câteva dintre opțiunile GCC pe care le puteți utiliza.

Tabelul 22-1 Opțiuni ale compilatorului GCC folosite uzual

Opțiune	Semnificație
<code>-ansi</code>	Poate lucra numai cu standardul ANSI C. (Această opțiune dezactivează câteva caracteristici specifice GNU C, ca de exemplu cuvintele cheie <code>asm</code> și <code>typeof</code> .)
<code>-c</code>	Compilează și generează numai obiecte fișier.
<code>-DMACRO</code>	Definește macro-ul, având valoarea de tip șir „1”.
<code>DMacro=DEFN</code>	Definește macro-ul ca DEFN.
<code>-E</code>	Rulează numai preprocesorul C.
<code>-fallow-single-precision</code>	Execută toate operațiile matematice în simplă precizie.
<code>-fpack-struct</code>	Împachetează toți membrii structurii fără a adăuga biți suplimentari.
<code>-fpcc-struct-return</code>	Returnează toate valorile <code>struct</code> și <code>union</code> în memorie, nu în regiștri. (Returnarea valorilor în acest fel este mai puțin eficientă, dar compatibilă cu alte compilatoare.)
<code>-fPIC</code>	Generează codul independent de poziție (position-independent code – PIC) potrivit adevărată o bibliotecă partajabilă.
<code>-freg-struct-return</code>	Returnează valorile <code>struct</code> și <code>union</code> în regiștri, atunci când este posibil.
<code>-g</code>	Generează informații de depanare. (Depanatorul GNU poate folosi această informație.)
<code>-IDIRECTORY</code>	Caută în directorul specificat fișiere incluse cu directiva preprocesor <code>#include</code>
<code>-LDIRECTORY</code>	Caută biblioteci în directorul specificat.
<code>-llibRARY</code>	Caută biblioteca specificată la realizarea legăturii.
<code>-m486</code>	Cod de optimizare pentru un PC 486. (Acest cod va rula și pe PC 386.)
<code>-o FILE</code>	Generează fișierul rezultat specificat (folosit pentru a desemna numele unui fișier executabil).
<code>-O0</code>	Nu optimizează.

(continuare)

Tabelul 22-1 Continuare

Opțiune	Semnificație
-O sau -O1	Optimizează codul generat.
-O2	Optimizează chiar suplimentar.
-O3	Realizează optimizări suplimentare față de cele realizate prin opțiunea -O2.
-pedantic	Generează erori dacă nu sunt folosite extensii ANSI standard.
-pg	Adaugă cod suplimentar programului, astfel încât atunci când este rulat, generează informații pe care programul gprof le poate folosi pentru a afișa detalii de timp pentru diferite părți ale programului.
-shared	Generează un fișier obiect partajat (folosit de obicei pentru a crea biblioteca partajată).
-traditional	Poate lucra numai cu sintaxa C tradițională Kernighan și Ritchie.
-DMACRO	Elimină definiția macro-ului specificat.
-v	Afișează numărul versiunii GCC.
-w	Nu generează nici un mesaj de avertizare.
-W1, OPTION	Transmite parametrul și OPTION (conținând opțiuni multiple, separate prin virgulă), programului ce realizează legarea. Pentru a crea o bibliotecă partajată numită libxxx.so.1, de exemplu, folosiți următorul indicator: -W1, -soname, libxxx.so.1

Utilitarul GNU make

Când o aplicație este alcătuită din mai multe fișiere sursă, compilarea și legarea acestora prin tastarea manuală a comenzii gcc nu mai este convenabilă. În plus, nu doriți să compilați fiecare fișier ori de câte ori modificați câte ceva într-un singur fișier sursă. Pentru această situație este folositor utilitarul GNU make.

Utilitarul make lucrează citind și interpretând un fișier makefile: un fișier text pe care trebuie să-l pregătiți conform unei sintaxe specificate. Fișierul makefile conține fișierele care constituie un program și prezintă modul de compilare și legare pentru a genera programul. Când modificați unul sau mai multe fișiere, make decide care din ele trebuie recompile și apoi emite comenzile corespunzătoare pentru compilare și re-generarea programului.



Utilitarul make este de fapt specificat în Secțiunea 6.2 a standardului POSIX.2 (IEEE Standard 1003.2-1992) pentru interfețe și instrumente. GNU make este în conformitate cu standardul POSIX.2

Numele fișierelor makefile

Prestabilit, utilitarul GNU make caută un fișier makefile cu unul din următoarele nume, în ordinea următoare:

- GNUmakefile
- makefile
- Makefile

În sistemele UNIX se folosește de obicei Makefile ca nume al fișierului *makefile*, deoarece el apare lângă începutul listei cu directoare, unde numele scrise cu majuscule apar înaintea celor scrise cu litere mici.

Când descărcați aplicații software din Internet, veți găsi de obicei un fișier Makefile împreună cu fișierele sursă. Pentru a genera programul, trebuie doar să tastați **make** la prompt-ul interpretorului; make se ocupă de toți etapele necesare generării programului.

Dacă fișierul dumneavoastră *makefile* nu are un nume standard, precum Makefile, trebuie să folosiți opțiunea -f pentru a-i specifica numele. Dacă fișierul dumneavoastră *makefile* este denumit *webprog.mak*, de exemplu, trebuie să lansați în execuție make cu următoarea linie de comandă:

```
make -f webprog.mak
```

GNU make acceptă, de asemenea, alte câteva opțiuni pentru linia de comandă, descrise în secțiunea „Rularea programului make” a acestui capitol.

Fișierul makefile

Pentru un program alcătuit din numeroase fișiere sursă și antet, fișierul *makefile* specifică următoarele:

- Articolele ce trebuie create cu make – de obicei fișierele obiect și cele executabile. Pentru un articol ce va fi creat se folosește termenul *țintă*
- Cum depind articolele (create cu make) de alte fișiere
- Ce comandă ar trebui executată pentru a crea fiecare țintă

Să presupunem că aveți un fișier sursă C++ denumit *form.C*, care conține următoarea directivă preprocesor:

```
#include „form.h” // Include header file
```

Fișierul obiect *form.o* depinde, evident, de fișierul sursă *form.C* și de fișierul antet *form.h*. Pe lângă aceste dependențe, trebuie să specificați cum ar trebui să convertească *make* fișierul *form.C* la fișierul obiect *form.o*. Să presupunem că doriți ca make să invoce g++ (deoarece fișierul sursă este în C++) cu următoarele opțiuni:

- -c (numai compilare)
- -g (generează informații de depanare)
- -O2 (face unele optimizări)

În fișierul *makefile*, puteți exprima aceasta cu următoarea regulă:

```
# Acesta este un comentariu în fișierul makefile
# Urmatoarele linii indica cum depinde form.o de
# form.C și form.h si cum sa creati form.o
```

```
form.o:form.C form.h
g++ -c -g -O2 form.C
```

În acest exemplu, prima linie care nu este comentariu indică *form.o* ca *țintă* și *form.C*, *form.h* ca fișiere *dependente*. Linia ce urmează celei cu dependențele prezintă construcția fișierului *țintă* din dependentele acestuia. Această linie ar trebui să înceapă cu un tabulator.

Avantajul folosirii make este acela că previne compilațiile care nu sunt necesare. La urma urmei, puteți invoca g++ (sau gcc) într-un script de interpretor pentru a compila și lega toate fișierele ce compun aplicația dumneavoastră, dar scriptul de interpretor va compila totul, chiar dacă compilațiile nu sunt necesare. Pe de altă parte, GNU make construiește un fișier *țintă*

numai dacă unul sau mai multe din dependentele sale s-a modificat de când a fost construit ultima dată fișierul țintă. `make` verifică această modificare examinând data ultimei modificări pentru fișierul țintă și pentru dependentele acestuia.



Observație

Observați că `make` consideră fișierul țintă ca numele unui scop ce trebuie îndeplinit; ținta nu trebuie să fie neapărat un fișier. Puteți avea o regulă de genul:

```
clean:
rm -f *.o
```

Această regulă specifică o țintă abstractă numită `clean` care nu are dependente. Această declarație de dependență ne indică faptul că pentru a face `clean`, GNU `make` ar trebui să invoce comanda `rm -f *.o`, care șterge toate fișierele cu extensia `.o` (adică fișierele obiect). Astfel, efectul creării țintei `clean` este ștergerea fișierelor obiect.

Variabile (sau macro-uri)

Pe lângă atribuția elementară de creare a fișierelor țintă din dependentele acestora, GNU `make` include multe caracteristici interesante care simplifică exprimarea dependențelor și a regulilor de construire a unei ținte din dependentele acesteia. Dacă trebuie să compilați un număr mare de fișiere C++ cu aceleași opțiuni GCC, de exemplu, tastarea opțiunilor pentru fiecare fișier este plictisitoare. Puteți evita acest lucru definind o variabilă sau un macro în fișierul `make` astfel:

```
# Definirea macro-ului pentru numele compilatorului
CXX= g++
```

```
# Definirea unui macro pentru indicatorii GCC
CXXFLAGS= -O2 -g -m486
```

```
# O regula pentru generarea unui fisier obiect
form.o: form.C form.h
$(CXX) -c $(CXXFLAGS) form.C
```

În acest exemplu, `CXX` și `CXXFLAGS` sunt variabile `make`. GNU `make` preferă să le denumească variabile, dar cele mai multe utilitare UNIX `make` le denumesc macro-uri.

Pentru a folosi o variabilă oriunde în cadrul fișierului `makefile`, începeți cu semnul dolarului (\$) urmat de variabila inclusă între paranteze. GNU `make` înlocuiește toate aparițiile unei variabile cu definiția ei; astfel, el înlocuiește toate aparițiile `$(CXXFLAGS)` cu șirul `-O2 -g -m486`.

GNU `make` conține câteva variabile predefinite, care au semnificații speciale. Tabelul 22-2 prezintă aceste variabile. Pe lângă variabilele din tabelul 22-2, GNU `make` consideră că toate variabilele de mediu sunt predefinite.

Tabelul 22-2 Câteva variabile predefinite în GNU `make`

Variabilă	Semnificație
<code>%</code>	Numele de membru pentru fișierele țintă arhivate. Dacă fișierul țintă este <code>libDisp.a(image.o)</code> , de exemplu, <code>%</code> este <code>image.o</code> și <code>@</code> este <code>libDisp.a</code> .
<code>*</code>	Numele fișierului țintă fără extensie
<code>+</code>	Numele tuturor fișierelor dependente cu dependente duplicate, listate în ordinea apariției lor
<code><</code>	Numele primului fișier dependent

Variabilă	Semnificație
<code>?</code>	Numele tuturor fișierelor dependente (cu spații între ele) mai recente decât fișierul țintă.
<code>@</code>	Numele complet al fișierului țintă.
<code>^</code>	Numele tuturor fișierelor dependente, cu spații între ele. Numele de fișiere dependente duplicate sunt îndepărtate
<code>AR</code>	Numele programului ce întreține arhiva. (Valoarea prestabilită: <code>ar</code> .)
<code>ARFLAGS</code>	Indicatori pentru programul ce întreține arhiva. (Valoarea prestabilită: <code>rv</code> .)
<code>AS</code>	Numele programului asamblor care convertește limbajul de asamblare în cod obiect. (Valoarea prestabilită: <code>as</code> .)
<code>ASFLAGS</code>	Indicatori pentru asamblor
<code>CC</code>	Numele compilatorului C. (Valoarea prestabilită: <code>cc</code> .)
<code>CFLAGS</code>	Indicatori ce trebuie transmiși compilatorului C
<code>CO</code>	Numele programului ce extrage un fișier din RCS. (Valoarea prestabilită: <code>co</code> .)
<code>COFLAGS</code>	Indicatori pentru programul RCS <code>co</code>
<code>CPP</code>	Numele preprocesorului C. (Valoarea prestabilită: <code>\$(CC) -E</code> .)
<code>CPPFLAGS</code>	Indicatori pentru preprocesorul C
<code>CXX</code>	Numele compilatorului C++. (Valoarea prestabilită: <code>g++</code> .)
<code>CXXFLAGS</code>	Indicatori ce trebuie furnizați compilatorului C++
<code>FC</code>	Numele compilatorului FORTRAN. (Valoarea prestabilită: <code>f77</code> .)
<code>FFLAGS</code>	Indicatori pentru compilatorul FORTRAN
<code>GET</code>	Numele programului care extrage un fișier din SCCS (Valoarea prestabilită: <code>get</code> .)
<code>GFLAGS</code>	Indicatori pentru programul SCCS <code>get</code>
<code>LDFLAGS</code>	Indicatori pentru compilator folosiți la invocarea programului de legare <code>ld</code>
<code>LEX</code>	Numele programului ce convertește gramatica Lex în program C. (Valoarea prestabilită: <code>lex</code> .)
<code>LFLAGS</code>	Indicatori pentru Lex
<code>MAKEINFO</code>	Numele programului ce convertește fișierele sursă Texinfo în fișiere <code>info</code> (Valoarea prestabilită: <code>makeinfo</code> .)
<code>RM</code>	Numele comenzii ce șterge un fișier (Valoarea prestabilită: <code>rm -f</code> .)
<code>TEX</code>	Numele programului care generează fișiere TeX DVI din fișiere sursă TeX (Valoarea prestabilită: <code>tex</code> .)
<code>TEXI2DVI</code>	Numele programului care generează fișiere TeX DVI din surse Texinfo (Valoarea prestabilită: <code>texi2dvi</code> .)
<code>YACC</code>	Numele programului ce convertește gramatica YACC în programe C (Valoarea prestabilită: <code>yacc -r</code> .)
<code>YFLAGS</code>	Indicatori pentru yacc

Reguli implicite

Utilitarul GNU make include, de asemenea, reguli interne sau implicite, care definesc crearea de fișiere țintă specifice din diferite dependențe. Un exemplu de regulă implicită este comanda pe care make ar trebui să o execute pentru a genera un fișier obiect dintr-un fișier sursă C.

Utilitarul GNU make suportă două tipuri de reguli implicite:

■ **Reguli sufix.** Regulile sufix reprezintă metoda învechită de a defini o regulă implicită pentru make. O regulă sufix definește cum se convertește un fișier cu o anumită extensie într-un fișier cu altă extensie. Fiecare regulă sufix este definită cu ținta indicând o pereche de sufixe (extensii de fișiere). Regula sufix pentru convertirea unui fișier .c (fișier sursă C) într-un fișier .o (obiect), de exemplu, poate fi scrisă astfel:

```
.c.o:
$(CC) $(CFLAGS) $(CPPFLAGS) -c -o $@ $<
```

Această regulă folosește variabilele predefinite CC, CFLAGS și CPPFLAGS. Pentru nume de fișiere, regula folosește variabilele \$@ (numele complet al țintei) și \$< (numele primului fișier dependent).

■ **Reguli de model.** Aceste reguli au întrebuințări mai diverse, deoarece puteți specifica reguli de dependență mai complexe. O regulă model arată că o regulă obișnuită, cu excepția semnului procent (%) care apare în numele țintei. Dependențele țintei folosesc de asemenea semnul procent (%) pentru a indica legătura lor cu numele țintei. Următoarea regulă model specifică cum se convertește un fișier X.c într-un fișier X.o:

```
%.o: %.c
$(CC) $(CFLAGS) $(CPPFLAGS) -c -o $@ $<
```

Utilitarul GNU make are un set considerabil de reguli implicite, definite atât ca reguli sufix cât și ca reguli model. Pentru a vedea o listă a tuturor regulilor și variabilelor, lansați în execuție make cu următoarea comandă:

```
make -p -f/dev/null
# GNU Make version 3.76.1, by Richard Stallman and Roland McGrath.
# Copyright (C) 1988, 89, 90, 91, 92, 93, 94, 95, 96, 97
# Free Software Foundation, Inc.
# Acesta este software gratuit; vezi sursa pentru condițiile copierii.
# Nu există NICI O garanție; nici măcar pentru COMERCIALIZARE sau pentru O
# DESTINAȚIE SPECIALĂ.
# Raportați erorile la adresa <bug-gnu-utils@prep.ai.mit.edu>.
```

```
make: *** No targets. Stop.
```

```
# Make data base, printed on Sat May 23 15:25:04 1998
# (linii sterse...)
```

```
# prestabilit
CC = cc
# prestabilit
OUTPUT_OPTIONS = -o $@
# prestabilit
COMPILE.c = $(CC) $(CFLAGS) $(CPPFLAGS) $(TARGET_ARCH) -c (lines deleted...)
```

```
# Reguli implicite
%.o: %.c
# Comenzi de executat (interne):
$(COMPILE.c) $< $(OUTPUT_OPTION)
# (linii sterse...)
```

Acest listing prezintă numai anumite părți selectate ale mesajului afișat de comanda GNU make, invocată cu opțiunea -p. Mesajul afișat conține numele variabilelor, precum și regulile implicite.

Un exemplu de fișier makefile

Puteți scrie ușor un fișier makefile dacă folosiți variabilele predefinite și regulile interne ale utilitarului GNU make. Să considerăm, de exemplu, un fișier makefile care creează executabilul xdraw din trei fișiere sursă C (xdraw.c, xviewobj.c și shapes.c) și două fișiere antet (xdraw.h și shapes.h). Să presupunem că fiecare fișier sursă conține unul din fișierele antet. Cu aceste considerații, un exemplu de fișier makefile poate arăta astfel:

```
#####
# Exemplu de fișier makefile
# Comentariile încep cu '#'
#
#####
```

```
# Folositi variabilele standard pentru a defini indicatori de compilare si
# legatura
```

```
CFLAGS= -g -O2
```

```
# Definirea targetului „all „
```

```
all: xdraw
```

```
OBJS=xdraw.o xviewobj.o shapes.o
```

```
xdraw: $(OBJS)
```

```
# Fișiere obiect
```

```
xdraw.o: Makefile xviewobj.c xdraw.h
```

```
shapes.o: Makefile shapes.c shapes.h
```

Acest fișier makefile se bazează pe regulile implicite ale utilitarului GNU make. Conversia fișierelor .c în fișiere .o folosește regula internă. Indicatorii sunt transmiși compilatorului C prin definirea variabilei CFLAGS.



Secret

Ținta all este definită prima din următorul motiv: dacă rulați GNU make fără specificarea unei ținte în linia de comandă (vezi sintaxa make descrisă în următoarea secțiune), el construiește prima țintă pe care o găsește în fișierul makefile. Definind prima țintă, all, ca xdraw, sunteți siguri că make construiește acest fișier executabil chiar dacă nu-l specificați explicit ca target. Programatorii în UNIX folosesc de obicei all ca nume al primei ținte, dar numele nu are importanță; contează numai faptul că este prima țintă din fișierul makefile.

Dacă aveți un director conținând fișierele sursă și fișierele antet adecvate, puteți încerca fișierul makefile. Iată ce se întâmplă când testez fișierul demonstrativ makefile:

```
make
cc -g -O2 -c xdraw.c -o xdraw.o
cc -g -O2 -c xviewobj.c -o xviewobj.o
cc -g -O2 -c shapes.c -o shapes.o
cc xdraw.o xviewobj.o shapes.o -o xdraw
```

Așa cum arată mesajul emis, make folosește comanda cc (care este o legătură simbolică cu GCC din sistemul dumneavoastră Linux) cu opțiunile corespunzătoare pentru a compila fișierele sursă și, în final, de a lega obiectele pentru a crea executabilul xdraw.

Rularea programului make

De obicei, lansați în execuție `make` cu o singură comandă, adică tastând `make`. Când este invocat în acest fel, utilitarul GNU `make` caută un fișier denumit `GNUmakefile`, `makefile` sau `Makefile` – în această ordine. Dacă `make` găsește unul din aceste fișiere `makefile`, el generează prima țintă specificată în acel `makefile`. Totuși, dacă `make` nu găsește un fișier `makefile` adecvat, afișează următorul mesaj de eroare și termină execuția:

```
make: *** No targets. Stop
```

Dacă fișierul dumneavoastră `makefile` are un nume diferit decât cele prestabilite, trebuie să folosiți opțiunea `-f` pentru a-l specifica. Sintaxa acestei opțiuni a comenzii `make` este:

```
make -f filename
```

unde `filename` este numele fișierului `makefile`.

Chiar atunci când aveți un fișier `makefile` cu un nume prestabilit precum `Makefile`, puteți genera o anumită țintă din cele definite în fișierul `makefile`. În acest caz, trebuie să rulați `make` cu următoarea sintaxă:

```
make target
```

Dacă fișierul `makefile` conține ținta numit `clean`, o puteți genera cu următoarea comandă:

```
make clean
```

Altă sintaxă specială suprascrie valoarea unei variabile `make`. De exemplu, GNU `make` folosește variabila `CFLAGS` pentru a păstra indicatorii folosiți la compilarea fișierelor C. Puteți suprascrie valoarea acestei variabile când invocați comanda `make`. Iată un exemplu care vă arată cum puteți defini variabila `CFLAGS`, astfel încât ea să reprezinte opțiunile `-g -O2`:

```
make CFLAGS="-g -O2"
```

Pe lângă aceste opțiuni, utilitarul GNU `make` mai acceptă și altele în linia de comandă. Tabelul 22-3 prezintă opțiunile utilitarului GNU `make`.

Tabelul 22-3 Opțiuni pentru utilitarul GNU make

Opțiune	Semnificație
-b	Ignorată, dar acceptată pentru compatibilitatea cu alte versiuni <code>make</code>
-C DIR	Intră în directorul specificat înaintea citirii fișierului <code>makefile</code>
-d	Tipărește informații de depanare
-e	Permite variabilelor de mediu să suprascrie definițiile variabilelor denumite identic, din fișierul <code>makefile</code> .
-f FILE	Citește FILE ca fișier <code>makefile</code>
-h	Afișează lista opțiunilor programului <code>make</code>
-i	Ignoră toate erorile ce apar în comenzile executate când se generează o țintă.
-I DIR	Caută în directorul specificat fișiere <code>makefile</code> (facilitatea de a include un fișier într-un fișier <code>makefile</code> este unică pentru GNU <code>make</code>)
-j NUM	Specifică numărul de comenzi <code>make</code> ce pot rula simultan
-k	Continuă să genereze ținte necorelate, chiar dacă intervine o eroare la generarea uneia din ele.

Opțiune	Semnificație
-l LOAD	Nu lansează o lucrare nouă dacă încărcarea medie nu este cel puțin LOAD (un număr în virgulă mobilă)
-m	Ignorată, dar acceptată pentru compatibilitatea cu alte versiuni <code>make</code>
-n	Tipărește comenzile ce trebuie executate, dar nu le execută
-o FILE	Nu regenerează fișierul numit FILE, chiar dacă este mai vechi decât dependentele sale
-p	Afișează baza de date cu variabile și reguli implicite ale utilitarului <code>make</code> .
-q	Nu rulează nimic, dar returnează zero dacă toate țintele sunt actualizate; returnează 1 dacă este necesară o actualizare și 2 dacă intervine o eroare.
-r	Elimină toate regulile interne.
-s	Lucrează în mod silențios (fără să afișeze comenzile pe măsură ce sunt executate)
-t	Schimbă marca de timp a fișierelor fără să le modifice propriu-zis
-v	Afișează numărul versiunii <code>make</code> și o notiță de copyright
-w	Afișează numele directorului de lucru înainte și după procesarea fișierului <code>makefile</code>
-W FILE	Consideră că fișierul specificat a fost modificat

Debanatorul GNU

Cu toate că `make` automatizează procesul generării unui program, nu este treaba dumneavoastră atunci când acesta nu rulează corect sau când întrerupe instantaneu rularea afișând un mesaj de eroare. Aveți nevoie de un debanator pentru a găsi cauza erorilor din program. CD-ROM-ul anexat acestei cărți include `gdb`, multifuncționalul debanator GNU cu o interfață în linie de comandă. CD-ROM-ul mai include `xxgdb`, o interfață grafică pentru `gdb` bazată pe X Window System.

Ca orice debanator, `gdb` vă permite să realizați depanări tipice, cum ar fi:

- Setarea unui punct de întrerupere (breakpoint) astfel încât execuția programului se oprește la o linie specificată.
- Urmărirea valorilor variabilelor din program
- Parcurgerea programului linie cu linie
- Modificarea variabilelor în încercarea de a repara erorile.

Debanatorul `gdb` poate depana programe C și C++.

Pregătirea unui program pentru depanare

Dacă doriți să depanați un program cu `gdb`, trebuie să vă asigurați că informațiile de depanare generate de compilator sunt plasate în fișierul executabil. Informația de depanare conține numele variabilelor din programul dumneavoastră și corespondența adreselor din fișierul executabil cu liniile de cod din fișierul sursă. `gdb` are nevoie de această informație pentru a-și îndeplini funcțiile, precum oprirea după executarea unei linii specificate a codului sursă.



Sugestie

Pentru a vă asigura că executabilul este pregătit pentru depanare, folosiți opțiunea `-g` cu GCC. Puteți face acest lucru definind variabila `CFLAGS` în fișierul `makefile` astfel:

```
CFLAGS= -g
```

Lansarea în execuție a depanatorului gdb

Modul cel mai uzual pentru depanarea unui program este rularea gdb folosind comanda: `gdb progname`.

unde *progname* este numele fișierului executabil al programului. După ce rulează, gdb afișează următorul mesaj și vă solicită să introduceți o comandă:

GDB este software gratuit și sunteți bine veniți să distribuiți copii ale acestuia cu anumite condiții; tastati „show copying” pentru a vedea condițiile. Nu există absolut nici o garanție pentru GDB; tastati „show warranty” pentru detalii.

GDB 4.16 (i386-redhat-linux), Copyright 1996 Free Software Foundation, Inc... (Gdb)

Pentru a vedea o listă a comenzilor gdb, tastati **help**, astfel:

(gdb) **help**

Lista claselor de comenzi

```
running - Program în execuție
stack - - Examinarea stivei
data - - Examinează datele
breakpoint - Oprirea programului într-un anumit punct
files - - Specificarea și examinarea fișierelor
status - - Întrebări despre stare
support - - Facilități oferite
user-defined - - Comenzi definite de utilizator
aliases - - Pseudonime ale altor comenzi
obscure - - Caracteristici necunoscute
internals - - Comenzi de întreținere
```

Tastati „help” urmat de numele clasei pentru a obține o listă a comenzilor din acea clasă.

Tastati „help” urmat de numele comenzii pentru a obține documentație completă. Abrevierile numelor de comenzi sunt permise dacă nu sunt ambigue. (Gdb)

Mesajul inițial afișează clasele de comenzi gdb. Puteți obține asistență suplimentară despre o anumită clasă de comenzi sau despre o anumită comandă dacă urmați instrucțiunile. Pentru a vedea lista de comenzi folosite pentru a rula programul pe care-l depanați, de exemplu, tastati `help running` la prompt-ul gdb.

Pentru a părăsi gdb, tastati **q** și apoi apăsați Enter.

gdb are un număr mare de comenzi, dar aveți nevoie numai de câteva pentru a găsi repede cauza unei erori. Tabelul 22-4 prezintă comenzile gdb cele mai utilizate.

Tabelul 22-4 Comenzi uzuale gdb

Comandă	Descriere
<code>break NUM</code>	Setează un punct de întrerupere la linia specificată (depanatorul se oprește la puncte de întrerupere).
<code>bt</code>	Afișează o istorie a tuturor cadrelor de stivă. (Această comandă vă arată succesiunea de apeluri până la momentul curent.)
<code>clear</code>	Șterge un punct de întrerupere de la o anumită linie dintr-un fișier sursă. <code>clear xdraw.c:8</code> , de exemplu, șterge punctul de întrerupere de la linia 8 a fișierului <code>xdraw.c</code> .
<code>FILENAME:NUM</code>	

Comandă	Descriere
<code>continue</code>	Continuă rularea programului ce este depanat. (Folosiți această comandă după ce programul s-a oprit datorită unui semnal sau al unui punct de întrerupere.)
<code>display EXPR</code>	Afișează valoarea expresiei (constând din variabile definite în program) de fiecare dată când programul se oprește.
<code>file FILE</code>	Încarcă fișierul executabil specificat pentru depanare.
<code>help NAME</code>	Afișează asistența pentru comanda NAME.
<code>info break</code>	Afișează o listă a punctelor de întrerupere curente, inclusiv de câte ori s-a ajuns la fiecare din acestea.
<code>info files</code>	Afișează informații detaliate despre fișierul ce este depanat.
<code>info func</code>	Afișează toate numele funcțiilor.
<code>info local</code>	Afișează informații despre variabilele locale ale funcției curente.
<code>info prog</code>	Afișează starea execuției programului ce este depanat.
<code>info var</code>	Afișează numele tuturor variabilelor globale și statistice.
<code>kill</code>	Încheie programul pe care-l depanați.
<code>list</code>	Listează o secțiune a codului sursă.
<code>make</code>	Rulează utilitarul make pentru a construi executabilul fără a ieși din programul gdb.
<code>next</code>	Avansează o linie de cod sursă în funcția curentă fără să treacă la alte funcții.
<code>print EXPR</code>	Arată valoarea expresiei EXPR.
<code>quit</code>	Părăsește gdb.
<code>run</code>	Începe rularea executabilelor încărcate în memorie.
<code>set variable VAR=VALUE</code>	Setează variabila VAR la valoarea VALUE.
<code>shell CMD</code>	Execută o comandă UNIX CMD fără să iasă din gdb.
<code>step</code>	Avansează o linie în funcția curentă, trecând la alte funcții dacă există.
<code>watch VAR</code>	Arată valoarea variabilei VAR la fiecare modificare a valorii acesteia.
<code>where</code>	Afișează succesiunea de apelare. Puteți folosi această comandă pentru a localiza unde s-a oprit programul dumneavoastră din cauza erorilor.
<code>x/F ADDR</code>	Examinează conținutul memoriei la adresa ADDR din formatul specificat de litera F, care poate fi o (octal), x (hexazecimal), d (zecimal), u (zecimal fără semn), t (binar), f (float), a (adresă), i (instrucțiune), c (char) sau s (string). La litera din format puteți adăuga o literă care indică mărimea tipului de date. Literele ce indică mărimea sunt b (băiți), h (halfword=jumătate de cuvânt, 2 băiți), w (word = cuvânt, 4 băiți) și g (giant=urias, 8 băiți). De obicei, ADDR este numele unei variabile sau al unui pointer.

Identificarea erorilor de programare cu gdb

Pentru a înțelege cum puteți găsi erori de programare cu gdb, cel mai ușor este să vă uitați la un exemplu, astfel că voi începe cu un program ipotetic ce conține o eroare de programare tipică.

Iată programul, pe care l-am stocat în fișierul `dbgtst.c`:


```
# include <stdio.h>

static char buf [256]
void read_input(char *s);

void main(void)
{
    char input = NULL; /* Doar un pointer, sirul nu va fi depozitat */

    read_input(input);

    /* Procesarea comenzii. */
    printf("You typed: %s\n", input);

    /* ... */
}

void read_input(char *s)
{
    printf("Command: ");
    gets (s);
}
```

Funcția main a acestui program apelează funcția read_input pentru a citi o linie introdusă de utilizator. Funcția read_input așteaptă să fie introdus un șir de caractere și returnează datele introduse de utilizator. În acest exemplu, funcția main apelează funcția read_input cu un pointer neinițializat, ceea ce constituie eroarea din acest program simplu.

Generați programul folosind gcc cu opțiunea -g, astfel:

```
gcc -g -o dbgtst dbgtst.c
```

Pentru a vedea problemele acestui program, lansați-l în execuție astfel:

```
dbgtst
Command: test
Segmentation fault
```

Programul este întrerupt și se afișează mesajul Segmentation fault (eroare de segmentare). Pentru acest mic program, puteți găsi cauza examinând codul sursă. Într-o aplicație autentică este posibil, însă, să nu depistați imediat ce a cauzat eroarea. De aceea, puteți folosi gdb pentru a găsi cauza problemei.

Pentru a localiza o eroare de programare folosind gdb, parcurgeți următorii pași:

1. Încărcați din gdb programul ce trebuie depanat. Pentru a încărca un program numit dbgtst în gdb, tastați următoarele:

```
gdb dbgtst
GDB is free software and you are welcome to distribute copies of it
under certain conditions; type „show copying” to see the conditions.
There is absolutely no warranty for GDB; type „show warranty” for details.
GDB 4.16 (i386-redhat-linux), Copyright 1996 Free Software
Foundation, Inc...
(gdb)
```

2. Rulați programul din gdb folosind comanda run. Dacă programul solicită să introduceți date, tastați textul de intrare. Programul ar trebui să se întrerupă, așa cum a făcut și mai înainte. Iată ce se întâmplă cu programul dbgtst:

```
(gdb)
Starting program: /home/naba/lxs2/test/dbgtst
```

```
Command: test
```

```
Program received signal SIGSEGV, Segmentation fault.
0x40046d10 in _IO_gets (buf=0x0) at iogets.c:62
iogets.c:62: No such file or directory.
(gdb)
```

3. Pentru a determina unde s-a întrerupt programul folosiți comanda where. Pentru programul dbgtst, această comandă afișează următorul mesaj:

```
(gdb) where
#0 0x40046d10 in _IO_gets (buf=0x0) at iogets.c:62
#1 0x80484f5 in read_input (s=0x0) at dbgtst.c:21
#2 0x80484c6 in main ( ) at dbgtst.c:10
(gdb)
```

Mesajul emis indică succesiunea apelării funcțiilor. Apelarea #0 – cea mai recentă – este a funcției gets din biblioteca C. Această funcție a fost apelată prima dată de funcția read_input, apelată, la rândul ei, de funcția main.

4. Folosiți comanda list pentru a inspecta liniile suspecte din codul sursă. Pentru programul dbgtst.c puteți începe cu linia 21, astfel:

```
(gdb) list dbgtst.c:21
16     }
17
18     void read_input(char *s)
19     {
20         printf("Command: ");
21         gets(s);
22     }
23
(gdb)
```

După ce ați examinat listingul anterior, vă veți da seama că problema constă în modul de apelare al funcției read_input. Apoi, veți lista liniile învecinate cu linia 10 din dbgtst.c (de unde a fost apelată funcția read_input), astfel:

```
(gdb) list dbgtst.c:10
5
6     void main(void)
7     {
8         char; / input = NUL; /* Doar un pointer, sirul nu va fi depozitat */
9
10        read_input(input);
11
12        /* Procesarea comenzii. */
13        printf("You typed: %s\n", input);
14
(gdb)
```

În acest punct veți observa că problema o reprezintă variabila denumită input.

Remediarea erorilor de programare cu gdb

Uneori, puteți încerca să remediați o eroare chiar în gdb. Pentru programul demonstrativ din secțiunea precedentă, puteți încerca această remediare imediat după ce programul este întrerupt datorită unei erori. Deoarece exemplul este ipotetic, am un buffer suplimentar numit buf definit în programul dbgtst, astfel:

```
static char buf[256];
```

Pot rezolva problema pointerului neinițializat setând variabila input pe valoarea buf. Următoarea sesiune cu gdb corectează problema pointerului neinițializat (acest exemplu este valabil imediat după ce programul a rulat și a fost întrerupt datorită erorii de segmentare):

```
(gdb) file dbgtst
Un program este deja în curs de depanare. Opriti depanarea? (y or n) y
```

```
Incarcati noua tabela de simboluri din „dbgtst”? (y or n) y
```

```
Citirea simbolurilor din dbgtst ...
```

```
done.
```

```
(gdb) list
```

```
1 #include <stdio.h>
```

```
2
```

```
3 static char buf [256]
```

```
4 void read_input(char *s);
```

```
5
```

```
6 void main(void)
```

```
7 {
8     char input = NULL; /* Just a pointer, no storage for string */
```

```
9
```

```
10 read_input(input);
```

```
(gdb) break 9
```

```
Breakpoint 1 at 0x80484bd: file dbgtst.c, line 9
```

```
(gdb) run
```

```
Starting program: /home/naba/1xs2/test/dbgtst
```

```
Breakpoint 1, main ( ) at dbgtst.c:10
```

```
10 read_input(input);
```

```
(gdb) set var input=buf
```

```
(gdb) cont
```

```
Continuing.
```

```
Command: test
```

```
You typed: test
```

```
Program exited with code 020.
```

```
(gdb)
```

Așa cum arată și lista de mai sus, dacă opresc programul chiar înainte ca funcția read_input să fie apelată și setez variabila denumită input pe buf (care este un șir de caractere valid), restul programului rulează corect.

După ce încercați un remediu care are efect, puteți face schimbările necesare în fișierele sursă și să permanentizați soluția găsită.

Implicațiile licențelor GNU

Trebuie să plătiți un preț pentru Linux – pentru a-și proteja programatorii și utilizatorii, Linux este distribuit cu GNU GPL (General Public License), care stipulează distribuirea codului sursă.

Acest lucru nu înseamnă că nu puteți scrie software comercial pentru Linux pe care să-l distribuiți (gratuit sau contra cost) numai în formă binară. Puteți să respectați toate regulile chiar dacă vindeți aplicații Linux în formă binară.

Când scrieți aplicații în Linux, ar trebui să țineți cont de următoarele două licențe:

- GNU General Public License (GPL) guvernează multe programe Linux, inclusiv nucleul Linux și GCC.

- GNU Library General Public License (LGPL) acoperă multe biblioteci Linux.



Atenție!

Următoarele secțiuni furnizează o imagine de ansamblu despre aceste licențe și câteva sugestii referitoare la îndeplinirea cerințelor acestor licențe. Deoarece eu nu sunt avocat, nimic din această carte nu trebuie considerat drept un sfat competent sub aspect juridic. După cum veți vedea, termenii integrali ai acestor licențe se găsesc în fișiere text din sistemul dumneavoastră Linux; arătați aceste licențe juristului dumneavoastră pentru o interpretare completă și o estimare a aplicabilității acestora în afacerea dumneavoastră

GNU General Public License (Licența publică generală)

Veți găsi conținutul licenței GPL într-un fișier numit COPYING în diferite directoare ale sistemului dumneavoastră Linux. De exemplu, tastează `cd /usr/doc/ghostscript*` și apoi more COPYING pentru a citi GPL.

GPL nu are nici o legătură cu distribuția gratuită sau contra cost a produselor software; scopul GPL este de a păstra programul gratuit pentru toți utilizatorii. GPL face acest lucru impunând ca programul să fie distribuit sub formă de cod sursă și stipulând că orice utilizator poate copia și distribui programul oricui sub formă de cod sursă. În plus, se reamintește tuturor că programul nu este livrat cu absolut nici o garanție.

Aplicațiile software care fac obiectul GPL nu sunt din domeniul public; astfel de aplicații software sunt întotdeauna protejate de legislația drepturilor de autor, iar GPL specifică restricțiile în ceea ce privește copierea și distribuirea acestora. Din punct de vedere al utilizatorului, bineînțeles, restricțiile GPL nu sunt chiar restricții; ele sunt avantaje, deoarece utilizatorului i se garantează accesul la codul sursă.



Atenție!

Dacă aplicația dumneavoastră folosește părți ale unor aplicații software ce fac obiectul GPL, atunci ea este considerată o operă derivată și intră sub incidența GPL, ceea ce înseamnă că trebuie să distribuiți codul sursă al aplicației.

Cu toate că nucleul Linux face obiectul GPL, această licență nu acoperă orice aplicație ce folosește serviciile nucleului prin apelarea funcțiilor sistem. Se consideră că acele aplicații reprezintă o utilizare normală a nucleului.

Dacă aveți de gând să distribuiți propria dumneavoastră aplicație în formă binară (așa cum sunt distribuite majoritatea aplicațiilor software comerciale), trebuie să vă asigurați că nu folosește nici o parte a unui software ce face obiectul GPL. Aplicația dumneavoastră poate ajunge să folosească părți din alt software atunci când apelează funcții dintr-o bibliotecă. Totuși, cele mai multe biblioteci fac obiectul altei licențe GNU, descrisă în secțiunea următoare.

Trebuie să fiți atent la câteva biblioteci și programe utilitare ce fac obiectul GPL. Baza de date cu bibliotecă a GNU dbm (gdbm) este una dintre bibliotecile remarcabile ce intră sub incidența GPL. Utilitarul analizor-generator GNU bison face, de asemenea, obiectul GPL. Dacă permiteți programului bison să genereze cod, atunci acel cod face obiectul GPL.



Secret

Pentru utilitarele GNU dbm și bison aveți alte alternative ce nu intră sub incidența GPL. Ca bază de date cu bibliotecă, puteți folosi baza de date Berkeley db în locul gdbm. Drept analizor-generator, puteți folosi yacc în loc de bison.

GNU Library General Public License (Licența publică generală pentru biblioteci)

Veți găsi conținutul licenței GNU LGPL într-un fișier numit `COPYING.LIB`. Dacă aveți nucleul sursă instalat, veți găsi o copie a fișierului `COPYING.LIB` într-unul din directorii sursă. Pentru a localiza o copie a fișierului `COPYING.LIB` folosiți următoarea comandă `find`:

```
find /usr -name "COPYING*" -print
```

Această comandă prezintă toate aparițiile fișierelor `COPYING` și `COPYING.LIB` din sistemul dumneavoastră. Fișierul `COPYING` conține GPL, în timp ce `COPYING.LIB` conține LGPL.

Scopul LGPL este de a permite folosirea bibliotecilor în aplicațiile dumneavoastră, chiar dacă nu distribuiți codul sursă al acestora. LGPL stipulează, totuși, că utilizatorii trebuie să aibă acces la codul sursă al bibliotecii pe care ați folosit-o și că aceștia pot folosi versiunile modificate ale acelor biblioteci.

Majoritatea bibliotecilor Linux, inclusiv biblioteca C (`libc.a`), fac obiectul LGPL. Astfel, când vă generați aplicația în Linux folosind compilatorul GCC, aceasta este legată împreună cu linii de cod de una sau mai multe biblioteci aflate sub incidența LGPL. Dacă vreți să vă distribuiți aplicația numai în forma binară, trebuie să fiți atenți la LGPL.



Secret



Veștile

O metodă pentru a satisface cerințele LGPL este de a fumiza codul obiect pentru aplicația și un fișier `makefile` care reface legătura între fișierele obiect și orice bibliotecă actualizată din Linux aflată sub incidența LGPL.

O modalitate mai bună de a satisface LGPL este folosirea *legării dinamice*, în care aplicația și biblioteca sunt entități separate, cu toate că aplicația dumneavoastră apelează funcții din bibliotecă atunci când rulează. Folosind legarea dinamică, utilizatorii vor beneficia imediat de orice actualizare a bibliotecilor, fără să fie necesar de a lega din nou aplicația.

Controlul versiunii

Când scrieți aplicații formate din câteva fișiere, este simplu să pregătiți un fișier `makefile` pentru automatizarea procesului de generare a programului și pentru a nu vă mai pierde timpul cu ținerea evidenței modificărilor. De obicei, pentru proiecte mici, puteți ține evidența modificărilor efectuate prin plasarea unor linii de comentariu la începutul fișierelor.

Dacă această metodă este bună pentru un proiect mic, pentru proiecte software mai mari se impune să folosiți unele instrumente care să vă ajute să gestionați diferite versiuni ale aplicațiilor dumneavoastră. De fapt, puteți beneficia de instrumente pentru controlul versiunii chiar dacă sunteți unicul autor al unei aplicații mici. După ceva timp, puteți să nu vă mai amintiți ce modificări ați făcut. Controlul versiunii programului vă poate ajuta să descifrați aceste modificări.

Distribuția software Linux este livrată cu RCS (Revision Control System – sistem pentru controlul reviziilor), care este o colecție de instrumente ce vă ajută să controlați versiunile revizuite ale programelor. Următoarele secțiuni vă oferă o imagine generală despre RCS și vă arată cum să-l folosiți prin intermediul câtorva exemple simple.

Instrumente pentru controlul sursei în RCS

Controlul sursei se referă la conceptul de salvare a versiunii codului sursă, astfel încât să puteți recupera o anumită versiune sau revizie a unui fișier oricând aveți nevoie. De fapt, atunci când modificați un fișier sursă se întâmplă următoarele:

1. Când aveți o versiune inițială a fișierului sursă, arhivați-o și plasați-o sub controlul sursei.
2. Când doriți să faceți vreo modificare în fișier, faceți mai întâi o copie a versiunii curente. (Când faceți acest lucru, instrumentele pentru controlul sursei trebuie să garanteze că nimeni nu poate modifica varianta respectivă.
3. Faceți modificările în fișier, testați codul și stocați fișierul modificat ca o versiune nouă.
4. Data viitoare când faceți modificări într-un fișier, începeți cu cea mai recentă variantă a acestuia.



Observație

RCS furnizează instrumente ce permit arhivarea reviziilor fișierelor și actualizarea lor într-un mod controlat. Tabelul 22-5 prezintă aceste instrumente.

Tabelul 22-5 Instrumente RCS

Instrument	Scop
<code>ci</code>	Creează o nouă variantă a unui fișier sau adaugă un fișier de lucru la un fișier RCS
<code>co</code>	Obține o versiune de lucru a unui fișier pentru citire. (<code>co -i</code> furnizează un fișier de lucru și blochează originalul, astfel încât să puteți modifica fișierul de lucru.)
<code>ident</code>	Caută identificatori într-un fișier
<code>merge</code>	Încorporează modificările din două fișiere într-un al treilea.
<code>rcsdiff</code>	Compară un fișier de lucru cu fișierul său RCS
<code>rcsmerge</code>	Unește diferite variante ale unui fișier
<code>rlog</code>	Vizualizează istoria modificărilor dintr-un fișier

RCS pentru începători

Să presupunem că tocmai ați terminat de programat versiunea inițială de lucru a unei aplicații și că doriți să folosiți RCS pentru a gestiona variantele viitoare. Următoarea secțiune descrie pașii pe care trebuie să-i urmați pentru a folosi RCS în munca dumneavoastră de programare.

Crearea fișierelor inițiale RCS

Primul pas în gestionarea variantelor modificate ale fișierelor sursă este să permiteți RCS să arhiveze varianta curentă a fișierelor dumneavoastră. Parcurgeți următorii pași pentru a pune fișierul sursă sub controlul RCS:

1. În directorul în care păstrați fișierele sursă ale aplicației dumneavoastră creați un subdirector denumit RCS, tastând `mkdir RCS`. Dacă subdirectorul RCS există, RCS arhivează variantele fișierelor în acesta.
2. În fiecare fișier pe care aveți de gând să-l plasați sub controlul variantei, adăugați un comentariu cu următorul cuvânt cheie de identificare RCS:

```
$Header$
```

Într-un fișier sursă C, de exemplu, adăugați următoarele:

```
/*
 * $Id$
 */
```

Pe de altă parte, într-un fișier `makefile` adăugați următoarele:

```
# $Id$
```

RCS detaliază mai târziu aceste cuvinte cheie de identificare pentru a include informații despre revizia fișierelor și data acestora.

3. Verificați fiecare fișier folosind comanda RCS și oferiți o scurtă descriere a fiecăruia, așa cum vă solicită comanda ci. Iată cum puteți verifica fișierul Makefile:

```
ci Makefile
RCS/Makefile,v < - - Makefile
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
> Makefile for sample programs .
>
initial revision: 1.1
Done
```

Când un fișier este verificat, comanda ci creează un fișier RCS corespunzător în directorul RCS. Numele fișierului RCS este identic cu numele fișierului original, cu excepția caracterelor „,v” adăugate la sfârșitul numelui. Astfel, fișierul RCS pentru fișierul Makefile este Makefile,v. De asemenea, ci șterge fișierul sursă original după ce creează fișierul RCS. Pentru a folosi sau edita fișierul din nou, trebuie să-l extrageți folosind comanda co.

După ce parcurgeți aceste etape, toate fișierele dumneavoastră vor fi stocate în siguranță în fișiere RCS din subdirectorul RCS.

Folosirea fișierelor arhivate

Să presupunem că doriți să editați unul din fișiere (de exemplu, să adăugați o nouă facilități) și să generați din nou aplicația. Ca începători, aveți nevoie de toate fișierele sursă și de fișierul makefile pentru etapa de compilare și legare.

Ar trebui să extrageți toate aceste fișiere folosind comanda co pentru acces numai la citire (read-only) (cu excepția fișierului pe care vreți să-l modificați, așa cum veți vedea imediat). Comanda co este ușor de folosit. Pentru a obține o copie de lucru a fișierului Makefile numai pentru citire, de exemplu, folosiți următoarea comandă:

```
co Makefile
RCS/Makefile,v - - > Makefile
revision 1.1
done
```

În mod prestabilit, această comandă caută un fișier RCS numit RCS/Makefile,v și creează o copie de lucru numai pentru citire, numită Makefile.

Veți găsi o copie a fișierului Makefile în director. Examinați-o și vedeți ce s-a întâmplat cu acel cuvânt cheie \$Id\$ pe care l-ați adăugat drept comentariu.

Iată cum arată fișierul meu Makefile demonstrativ:

```
# $Id: Makefile,v 1.1 1998/05/24 08:19:07 naba Exp $
```

Așa cum o arată acest exemplu, RCS detaliază fiecare cuvânt cheie identificator într-un șir cu informații. Aceste informații depind de identificator.

Dacă doriți să modificați un fișier, trebuie să-l verificați cu opțiunea -l. Dacă doriți să verificați o copie a fișierului xmutil.c, având acces la scriere, folosiți următoarea comandă:

```
co -l xmutil.c
RCS/xmutil.c,v - - > xmutil.c
revision 1.1 (locked)
don
```

Dacă comparați acest mesaj emis cu cel din exemplul anterior al comenzii co, veți observa că mesajul curent confirmă faptul că fișierul RCS este blocat. Nimeni nu poate modifica fișierul arhivat până nu terminați de verificat copia acestuia, având acces la scriere.

Când verificați un fișier și îl blocați, nimeni nu mai poate verifica același fișier, având acces la scriere. Totuși, oricine poate examina o copie a fișierului, destinată numai pentru citire.

După ce faceți modificări într-un fișier, puteți să-l verificați din nou folosind comanda ci, exact cum ați procedat la crearea fișierului RCS.

Folosirea cuvintelor cheie de identificare

Puteți folosi cuvintele cheie de identificare RCS – fiecare din ele este un șir delimitat în ambele părți de semnul dolar (\$...\$) – pentru a înregistra informații într-un fișier sursă. Sistemul RCS detaliază cuvântul cheie \$Id\$, de exemplu, într-o scurtă descriere a fișierului, inclusiv numele acestuia, numărul variantei, data și autorul. Trebuie doar să puneți cuvântul cheie în fișier; RCS are grijă să detalieze acel cuvânt cheie în informația adecvată. Tabelul 22-6 listează cuvintele cheie de identificare pe care le poate utiliza RCS.

Tabelul 22-6 Cuvinte cheie de identificare admise de sistemul RCS

Cuvânt cheie	Scop
\$Author\$	Identificatorul de conectare al utilizatorului care a verificat varianta.
\$Date\$	Data și ora când a fost verificată varianta
\$Header\$	Prezintă în informații pe scurt, inclusiv calea completă a fișierului RCS, numărul variantei, data, autorul și starea variantei fișierului
\$Id\$	La fel ca și \$Header\$, cu excepția că numele fișierului RCS nu are nici un director ca prefix
\$Locker\$	Identificatorul de conectare al utilizatorului care a blocat fișierul (vid dacă fișierul nu este blocat la momentul respectiv)
\$Log\$	Prezintă la un jurnal de modificări făcute în fișier
\$RCSfile\$	Numele fișierului RCS fără numele directoarelor
\$Revision\$	Numărul reviziei pentru fișierul RCS
\$Source\$	Detaliază la calea completă a fișierului RCS
\$State\$	Indică starea reviziei fișierului (dacă este blocat sau nu)



Sugestie



Este necesar să folosiți cel puțin cuvântul cheie \$Id\$ în fișierele dumneavoastră pentru a include informații pe scurt despre cea mai recentă variantă.

RCS detaliază cuvintele cheie identificatoare oriunde în fișier. Astfel, puteți însemna un fișier obiect (și executabilul ce folosește acel fișier obiect) plasând cuvântul cheie identificator într-o variabilă de tip șir. De obicei, se definește un șir numit rcsid astfel:

```
static const char rcsid [ ] =
    „$Id”;
```

Definirea șirului rcsid determină fișierul obiect și pe cel executabil să conțină un șir precum următorul:

```
$Id: xmutil.c,v 1.2 1998/05/24 08:31:20 naba Exp $
```

Alte comenzi RCS

De cele mai multe ori, veți folosi comenzile `ci` și `co` pentru a întreține reviziile fișierelor folosind RCS. Totuși, RCS include alte câteva instrumente pentru administrarea diferitelor aspecte ale controlului versiunii, precum compararea a două revizii, vizualizarea istoriei modificărilor și examinarea identificatorilor din fișiere.

Vizualizarea modificărilor făcute până la un moment dat

Când verificați modificările dintr-un fișier, doriți probabil să știți ce schimbări ați făcut până în prezent. Puteți folosi programul `rcsdiff` pentru a vedea o listă a modificărilor. Dacă editați un fișier numit `xmutil.c`, de exemplu, puteți compara fișierul de lucru cu fișierul sau RCS utilizând următoarea comandă:

```
rcsdiff xmutil.c
```

Programul `rcsdiff` lansează în execuție utilitarul Linux `diff` pentru a găsi diferențele între fișierul de lucru și fișierul RCS.

Dacă este necesar, puteți chiar să găsiți diferențele între două versiuni anume ale unui fișier, cu următoarea comandă:

```
rcsdiff -r1.1 -r1.2 xmutil.c
```

Această comandă listează diferențele între versiunile 1.1 și 1.2 ale fișierului `xmutil.c`.

Anularea anumitor modificări operate până la un moment dat

Uneori, după ce operați câteva modificări într-un fișier, vă dați seama că acestea sunt ori greșite, ori nu sunt necesare. În acest caz, doriți să renunțați la ele.

Pentru a renunța la modificări, trebuie doar să deblocați fișierul RCS și apoi să ștergeți copia de lucru a fișierului. Pentru a debloca un fișier RCS, folosiți comanda `rcs` cu opțiunea `-u` (unlock – deblocare). Următoarea comandă elimină modificările curente din fișierul numit `xmutil.c`:

```
rcs -u xmutil.c
```

Altă modalitate, mai convenabilă, de a elimina modificările este de a suprascrie fișierul de lucru curent cu o copie a vechiului fișier RCS. Pentru a face acest lucru, folosiți comanda `co` cu indicatorii `-u` și `-f`, astfel:

```
co -f -u xmutil.c
```

Opțiunea `-u` deblochează versiunea verificată iar `-f` forțează `co` să suprascrie fișierul de lucru cu revizia mai veche a acestuia.

Vizualizarea jurnalului de modificări

Pe măsură ce faceți modificări într-un fișier și verificați versiunile, RCS păstrează un jurnal al acestora. Puteți vizualiza acest jurnal folosind comanda `rlog`. Următorul exemplu arată cum să vizualizați jurnalul modificărilor pentru fișierul `xmutil.c`:

```
rlog xmutil.c
```

```
RCS file: RCS/xmutil.c,v
Working file: xmutil.c
head: 1.2
```

```
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2; selected revisions: 2
description:
Utility functions
-----
revision 1.2
date: 1998/05/26 06:31:20; author: naba; state: Exp; lines: +15 -1
minor revision
-----
revision 1.1
date: 1998/05/26 05:55:23; author: naba; state: Exp;
Initial revision
=====
```

Prima parte a ieșirii generate de comanda `rlog` afișează câteva informații pe scurt despre fișierul RCS. Ceea ce urmează după linia `description`: prezintă descrierea pe care am introdus-o atunci când am creat fișierul RCS. Veți vedea o înregistrare pentru fiecare revizie, cele mai recente apărând la început. Fiecare înregistrare a unei revizii prezintă data, autorul și o scurtă descriere dată de acesta.

Examinarea cuvintelor cheie de identificare

Dacă aveți vreun cuvânt cheie de identificare înglobat într-un fișier, puteți să-l vizualizați folosind comanda `ident`. Dacă fișierul `Makefile` conține cuvântul cheie `$Id$`, îl puteți examina cu următoarea comandă:

```
ident Makefile
```

```
Makefile:
```

```
$Id: Makefile,v 1.2 1998/05/24 08:25:47 naba Exp $
```

Dacă definiți o variabilă de tip `șir` cu un cuvânt cheie care este în cele din urmă înglobat într-un fișier binar (un fișier obiect sau unul executabil), puteți folosi, de asemenea, comanda `ident` pentru a vizualiza acei identificatori. Puteți încerca comanda `ident` cu orice fișier binar pentru a vedea dacă are înglobate cuvinte cheie. Iată ce am găsit când am încercat comanda `ident` cu fișierul `/usr/bin/co` (programul executabil al comenzii `co`):

```
ident /usr/bin/co
```

```
/usr/bin/co:
```

```
$Id: rcbase.h,v 5.20 1995/06/16 06:19:24 eggert Exp $
$Id: co.c,v 5.18 1995/06/16 06:19:24 eggert Exp $
$Id: rcslx.c,v 5.19 1995/06/16 06:19:24 eggert Exp $
$Id: rcssyn.c,v 5.15 1995/06/16 06:19:24 eggert Exp $
$Id: rcsgen.c,v 5.16 1995/06/16 06:19:24 eggert Exp $
$Id: rcsedit.c,v 5.19 1995/06/16 06:19:24 eggert Exp $
$Id: rcskeys.c,v 5.4 1995/06/16 06:19:24 eggert Exp $
$Id: rcsmmap.c,v 5.3 1995/06/16 06:19:24 eggert Exp $
$Id: rcsrev.c,v 5.10 1995/06/16 06:19:24 eggert Exp $
$Id: rcsutil.c,v 5.20 1995/06/16 06:19:24 eggert Exp $
$Id: rcsfnms.c,v 5.16 1995/06/16 06:19:24 eggert Exp $
$Id: maketime.c,v 5.11 1995/06/16 06:19:24 eggert Exp $
$Id: partime.c,v 5.13 1995/06/16 06:19:24 eggert Exp $
$Id: rcstime.c,v 1.4 1995/06/16 06:19:24 eggert Exp $
$Id: rcskeep.c,v 5.10 1995/06/16 06:19:24 eggert Exp $
```

Din acest mesaj rezultat, puteți spune exact care sunt versiunile fișierelor sursă ce au fost folosite pentru a construi această versiune a programului `co`.

Subiecte de programare în Linux

Programarea aplicațiilor în Linux este asemănătoare cu programarea în orice sistem UNIX. Cele mai multe subiecte de programare în C și UNIX sunt generale și se aplică tuturor sistemelor UNIX. Există, totuși, câteva subiecte pe care doriți să le cunoașteți dacă dezvoltați aplicații software în Linux.

Această secțiune tratează cel mai semnificativ subiect: noul format executabil și de legătură (Executable and Linking Format – ELF) binar în Linux. Celălalt subiect, folosirea legării dinamice în aplicații, este corelat cu ELF. De asemenea, este prezentată legarea dinamică și modul de creare a unei biblioteci cu legare dinamică în Linux.

Formatul executabil și de legare (ELF)

Dacă ați programat în Linux, probabil știți că atunci când compilați și legați un program, fișierul executabil prestabilit este `a.out`. Ceea ce probabil nu ați observat este că fișierul `a.out` este asociat cu un format de fișier. Sistemul de operare trebuie să cunoască acest format, astfel încât să poată încărca și rula un executabil. Linux folosește formatul `a.out` pentru fișierele binare încă de la începuturile lui.

Cu toate că formatul `a.out` și-a îndeplinit misiunea corespunzător, el are două dezavantaje:

- Bibliotecile partajate sunt dificil de creat.
- Încărcarea dinamică a unei biblioteci partajate este dificilă.

Folosirea bibliotecilor partajate este de dorit, deoarece o astfel de bibliotecă permite mai multor fișiere executabile să folosească același bloc de cod. De asemenea, încărcarea dinamică a modulelor devine din ce în ce mai populară, deoarece permite unei aplicații să încarce blocuri de cod numai atunci când este nevoie, reducând astfel cerințele de memorie ale aplicației.



Între timp, firma UNIX System Laboratories a dezvoltat un nou format binar, numit format executabil și de legare (Executable and Linking Format – ELF) pentru folosirea în System V Release 4 (SVR4). Formatul ELF este mult mai flexibil decât formatul `a.out`. În particular, ELF are următoarele avantaje față de formatul `a.out` în Linux:

- Bibliotecile partajate pentru formatul ELF sunt mai simplu de creat. Compilați toate fișierele sursă folosind comanda `gcc -fPIC -c` și apoi le legați cu o comandă, care creează biblioteca `libXXX.so.1.0 *.o`, cum este următoarea:
`gcc -shared -Wl, -soname, libXXX.so.1 -o libXXX.so.1.0 *.o`
- Încărcarea dinamică (în care un program încarcă module de cod în timpul rulării) este mai simplă. Cu încărcarea dinamică, puteți proiecta o aplicație care să fie extensibilă, astfel încât utilizatorii pot adăuga cod nou sub formă de biblioteci partajate.

Datorită flexibilității crescute a formatului ELF, programatorii Linux (în particular, programatorii GCC) au hotărât ca acesta să fie formatul binar standard în Linux. În mod prestabilit, noul compilator GCC – gcc Versiunea 2.7 – generează fișiere binare în format ELF.



Secret

Observați că GCC Versiunea 2.7 continuă să folosească `a.out` ca nume prestabilit al fișierului executabil (folosit numai dacă nu specificați un nume de fișier rezultat cu opțiunea `-o`). Cu toate că executabilul poate fi denumit `a.out`, formatul binar este ELF și nu vechiul format `a.out`.



Dacă doriți să modificați formatul binar al unui fișier executabil, folosiți comanda `file`. Următorul exemplu vă arată cum să verificați tipul fișierului `/bin/ls` (fișierul executabil pentru comanda `ls`):

```
file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1,
dynamically linked, stripped
```

Pe de altă parte, comanda `file` raportează pentru un format executabil `a.out` mai vechi următoarele:

```
file a.out
a.out: Linux/i386 demand-paged executable (QMAGIC)
```

Biblioteci partajate în aplicațiile Linux

Cele mai multe programe Linux folosesc biblioteci partajate. Cele mai multe programe C folosesc cel puțin biblioteca partajată `libc.so.x` din C, în care `x` este numărul versiunii. Când un program folosește una sau mai multe biblioteci partajate, pentru a rula programul aveți nevoie de fișierul executabil al programului precum și de toate bibliotecile partajate. Cu alte cuvinte, programul dumneavoastră va rula numai dacă toate bibliotecile partajate sunt disponibile în sistem.

Dacă vindeți o aplicație, trebuie să vă asigurați că toate bibliotecile partajate necesare sunt distribuite cu programul dumneavoastră

Examinarea bibliotecilor partajate ale unui program

Folosiți utilitarul `ldd` pentru a determina care sunt bibliotecile partajate necesare unui program executabil. Iată ce raportează `ldd` pentru un program C tipic compilat cu gcc Versiunea 2.7.2. (programul folosește formatul binar ELF):

```
ldd a.out
libc.so.6 => /lib/libc.so.6 (0x40003000)
lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00000000)
```

Pentru un program mai complex cum este `xv` (versiunea ELF), `ldd` afișează mai multe biblioteci partajate, astfel:

```
ldd /usr/X11/bin/xv
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40003000)
libjpeg.so.6 => /usr/lib/libjpeg.so.6 (0x4009a000)
libpng.so.0 => /usr/lib/libpng.so.0 (0x400b8000)
libz.so.1 => /usr/lib/libz.so.1 (0x400d1000)
libm.so.6 => /lib/libm.so.6 (0x400de000)
libc.so.6 => /lib/libc.so.6 (0x400f7000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00000000)
```

În acest caz, programul folosește câteva biblioteci partajate printre care biblioteca `X11` (`libX11.so.6`), biblioteca `JPEG` (`libjpeg.so.6`), biblioteca `Portable Network Graphics` (`PNG`) (`libpng.so.0`), biblioteca `Math` (`libm.so.6`) și biblioteca `C` (`libc.so.6`).

Astfel, aproape orice aplicație Linux are nevoie de biblioteci partajate pentru a rula. În plus, bibliotecile partajate trebuie să aibă același format binar ca și cel folosit de aplicație.

Crearea unei biblioteci partajate

Cu ELF, crearea unei biblioteci partajate pentru propria dumneavoastră aplicație este destul de simplă. Să presupunem că doriți să implementați un obiect sub forma unei biblioteci partajate.

Un set de funcții în biblioteca partajată va reprezenta interfața obiectului. Pentru a folosi obiectul, încărcați biblioteca lui partajată și apoi invocați funcțiile lui de interfață (veți învăța să faceți acest lucru în secțiunea următoare).

Iată codul sursă C pentru acest obiect simplu, implementat ca bibliotecă partajată (o mai puteți denumi și bibliotecă cu legare dinamică):

```
/*----- */
/* Fisier: dynobj.c
*
* Demonstreaza utilizarea legaturii dinamice.
* Pretinde ca acesta este un obiect ce poate fi creat apeland init si
* distrus apeland destroy.
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* Structura de date pentru acest obiect */
typedef struct OBJDATA
{
    char *name;
    int version;
} OBJDATA;
/*----- */
/* i n i t
*
* Initializarea obiectului (alocare de memorie).
*
*/
void* init(char *name)
{
    OBJDATA data = (OBJDATA*)calloc(1, sizeof (OBJDATA));
    if (name)
        data->name = malloc(strlen(name)+1);
    strcpy(data->name, name);

    printf("Created: %s\n", name);

    return data;
}
/*----- */
/* s h o w
*
* Arata obiectul.
*
*/
void show(void *data)
{
    OBJDATA *d = (OBJDATA*) data;
    printf("show: %s\n", d->name);
}
/*----- */
/* d e s t r o y
*
* Distruge obiectul (elibereaza toata memoria).
```

```
*
*/
void destroy(void *data)
{
    OBJDATA *d = (OBJDATA*) data;
    if(d)
    {
        if (d->name)
        {
            printf("Destroying: %s\n", d->name);
            free(d->name);
        }
        free(d);
    }
}
```

Obiectul oferă trei funcții de interfață:

- init pentru alocarea unui spațiu de stocare necesar și inițializarea obiectului
- show pentru afișarea obiectului (în acest caz, pur și simplu afișează un mesaj)
- destroy pentru a elibera toate spațiile de stocare

Pentru a genera biblioteca partajată numită libdobj.so, parcurgeți următorii pași:

1. Compilați toate fișierele sursă cu indicatorul `-fPIC`. În acest caz, compilați fișierul `dynobj.c` cu următoarea comandă:

```
gcc -fPIC -c dynobj.c
```

2. Legați obiectele într-o bibliotecă partajată cu indicatorul `-shared` și furnizați indicatorii adecvați pentru programul care realizează legarea. Pentru a crea biblioteca partajată numită `libdobj.so.1` folosiți următoarele:

```
gcc -shared -Wl, -soname, libdobj.so.1 -o libdobj.so.1.0 dynobj.o
```

3. Creați o succesiune de legături simbolice, astfel încât programele ce folosesc biblioteca partajată să poată face referire la aceasta folosind un nume standard. Pentru biblioteca dată ca exemplu, numele standard este `libdobj.so` și legăturile simbolice sunt create cu următoarele comenzi:

```
ln -s libdobj.so.1.0 libdobj.so.1
ln -s libdobj.so.1 libdobj.so
```

4. Când testați biblioteca partajată, definiți și exportați variabila de mediu `LD_LIBRARY_PATH` cu următoarea comandă:

```
export LD_LIBRARY_PATH= pwd :$LD_LIBRARY_PATH
```

După ce testați biblioteca partajată și sunteți satisfăcuți că aceasta funcționează, se recomandă să o copiați într-o locație standard, ca de exemplu `/usr/local/lib`, și să rulați utilitarul `ldconfig` pentru a actualiza legătură dintre `libdobj.so.1` și `libdobj.so.1.0`. Iată care sunt comenzile pentru instalarea bibliotecii dumneavoastră partajate pentru uzul general (trebuie să intrați în sistem ca root pentru a executa aceste etape):

```
cp s libdobj.so.1.0 /usr/local/lib
/sbin/ldconfig
cd /usr/local/lib
ln -s libdobj.so.1 libdobj.so
```


Încărcarea dinamică a unei biblioteci partajate

ELF simplifică încărcarea unei biblioteci partajate în programul dumneavoastră și folosirea funcțiilor din cadrul acesteia. Fișierul antet <dlfcn.h> declară funcțiile pentru încărcarea și folosirea unei biblioteci partajate. Patru funcții sunt declarate în fișierul dlfcn.h pentru încărcare dinamică:

- `void *dlopen(const char *filename, int flag);` Încarcă biblioteca partajată specificată de filename și returnează un mâner (handle) al acesteia. Indicatorul (flag) poate fi `RTD_LAZY` (stabilește simbolurile nedefinite pe măsură ce codul bibliotecii este executat sau `RTD_NOW` (stabilește toate simbolurile nedefinite înainte de terminarea execuției funcției `dlopen` și eșuează dacă toate simbolurile nu sunt definite). Dacă `dlopen` nu reușește, returnează `NULL`.
- `const char *dlerror (void);` Dacă `dlopen` nu funcționează, apelați `dlerror` pentru a obține un șir ce conține o descriere a erorii.
- `void *dlsym (void *handle, char *symbol);` Returnează adresa simbolului specificat (numele funcției) din biblioteca partajată identificată prin maner (care a fost returnat de `dlopen`).
- `int dlclose (void *handle);` Descarcă biblioteca partajată dacă n-o folosește nimeni.

Când folosiți oricare din aceste funcții, includeți fișierul antet <dlfcn.h> cu următoarea directivă preprocesor:

```
#include <dlfcn.h>
```

În continuare este prezentat un program test simplu, care arată cum să încărcați și să folosiți obiectul definit în biblioteca partajată `libdobj.so`, pe care ați creat-o în secțiunea precedentă:

```
/*-----*/
/* Fisier: dltest.c
 *
 * Testarea legaturii dinamice.
 *
# include <dlfcn.h> /* Pentru incarcarea dinamica a functiilor */
# include <stdio.h>

void main (void)
{
    void *dlobj;
    void * (*init_call) (char *name);
    void (*show_call) (void *data);
    void (*destroy_call) (void *data);

    /* Deschide biblioteca partajata si seteaza pointerii functiei */
    if (dlobj = dlopen ("libdobj.so.1",RTLD_LAZY)
    {
        void *data;

        init_call=dlsym (dlobj,"init");
        show_call=dlsym (dlobj," show");
        destroy_call=dlsym (dlobj," destroy");

    /* Apeleaza interfetele obiectului */
        data = (*init_call) ("Test Object");
        (*show_call) (data);
        (*destroy_call) (data);
    }
}
```

Programul este simplu: încarcă biblioteca partajată, detectează pointerii funcțiilor din bibliotecă și apelează funcțiile prin intermediul pointerilor.

Puteți compila și lega acest program în mod obișnuit, cu excepția faptului că trebuie să realizați legarea cu opțiunea `-ldl` pentru a putea folosi funcțiile declarate în fișierul antet <dlfcn.h>.

Iată cum trebuie să generați programul `dltest`:

```
gcc -o dltest dltest.c -ldl
```

Pentru a vedea programul în acțiune, rulați `dltest` astfel:

```
dltest
```

```
Created: Test Object
```

```
show: Test Object
```

```
Destroying: Test Object
```

Cu toate că această procedură nu pare interesantă, aveți acum o mostră de program ce folosește o bibliotecă partajată.

Pentru a vedea avantajele folosirii unei biblioteci partajate, mergeți la secțiunea precedentă și faceți câteva modificări în biblioteca partajată (de exemplu, setați o anumită funcție să afișeze alt mesaj). Regenerați numai biblioteca partajată și rulați din nou `dltest`. Mesajul rezultat trebuie să reflecte efectul schimbărilor operate în biblioteca partajată, ceea ce înseamnă că puteți actualiza biblioteca independent de aplicație.

Rezumat

Sistemul dumneavoastră Linux este livrat cu toate instrumentele necesare dezvoltării aplicațiilor software. De fapt, veți găsi toate instrumentele de dezvoltare software GNU, precum GCC, depanatorul GNU, utilitarul GNU `make` și utilitarul pentru controlul versiunii RCS. Capitolul de față descrie aceste instrumente de dezvoltare software și vă arată cum să le folosiți. Din lectura acestui capitol, ați învățat următoarele:

- ▶ Instrumentele GNU includ mediul de dezvoltare software de pe PC-ul dumneavoastră Linux. Aceste instrumente includ GNU Emacs pentru editarea textului, GCC pentru compilarea programelor C și C++, utilitarul GNU `make` pentru automatizarea generării aplicațiilor, depanatorul GNU și RCS pentru controlul versiunii.
- ▶ Un utilitar denumit `info` furnizează informații de asistență online despre instrumentele GNU. Puteți rula `info` separat sau din cadrul editorului GNU Emacs folosind comanda `Ctrl+h i`.
- ▶ GCC este compilatorul GNU pentru C și C++. Puteți folosi comanda `gcc` pentru a compila și lega programe C. Folosiți `g++` pentru a compila și lega programe C++.
- ▶ Compilatorul GCC are multe opțiuni, dar numai câteva sunt necesare. Printre cele mai folosite sunt `-c` (numai pentru compilare) și `-o` (pentru specificarea numelui fișierului executabil rezultat).
- ▶ Utilitarul GNU `make` vă permite să automatizați procesul de generare a aplicațiilor. Specificați modulele ce alcătuiesc un executabil precum și toate fișierele dependente; `make` are grijă să compileze numai acele fișiere ce necesită recompilare. Fișierul de intrare pentru `make` este cunoscut ca `makefile` și este denumit uzual `Makefile`.
- ▶ Depanatorul GNU vă permite să localizați erorile din programul dumneavoastră. Folosiți comanda `gdb` pentru a lansa în execuție depanatorul. Trebuie să compilați programul cu opțiunea `-g` a programului GCC pentru a genera informații de depanare pe care depanatorul GNU le poate folosi.

- ▶ Când folosiți instrumente GNU pentru dezvoltarea aplicațiilor (așa cum veți face în Linux), ar trebui să țineți cont de licențele GNU: GNU General Public License (GPL) și GNU Library General Public License (LGPL). Bibliotecile GNU se află sub incidența LGPL. Dacă distribuiți programul sub formă binară, trebuie să folosiți legarea dinamică pentru a respecta prevederile LGPL. De asemenea, nu considerați nici o informație din această carte ca fiind un sfat competent de natură juridică; întotdeauna consultați-vă juristul pentru un răspuns fondat.
- ▶ Controlul versiunii este un aspect important al programării. În Linux, aveți la dispoziție RCS (Revision Control System) pentru a gestiona versiunile fișierelor sursă. Cu RCS puteți folosi comenzi precum `ci` și `co` pentru a gestiona fișierele sursă. Puteți îngloba siruri precum `$Id$` sau `$header$` pentru a marca versiunea și data în fișierele dumneavoastră sursă.
- ▶ Bibliotecile partajate sunt folosite uzual în aplicațiile Linux pentru a reduce necesarul de memorie al fișierelor executabile. Aplicațiile sunt legate dinamic cu o bibliotecă partajată în timpul rulării și multe aplicații pot folosi o singură bibliotecă partajată.
- ▶ Comunitatea programatorilor în Linux a adoptat formatul ELF (Executable and Linking Format) pentru fișiere binare. ELF realizează legarea dinamică a programului într-un mod mai simplu. Acest capitol vă prezintă modul de utilizare a legăturii dinamice în propriile dumneavoastră aplicații.

Capitolul 23

Programarea X în Linux

În acest capitol

- ▶ Veți înțelege programarea condusă prin evenimente
- ▶ Veți înțelege funcțiile pentru apel invers (callback) și subrutinele de tratare a evenimentelor
- ▶ Veți scrie un program Motif
- ▶ Veți înțelege primitivele (widgets) Motif
- ▶ Veți seta resursele pentru primitive
- ▶ Veți folosi funcțiile Xlib în programele Motif
- ▶ Veți rezuma evenimentele X
- ▶ Veți include text, grafică și imagini în programele Motif

Capitolul 22 v-a arătat cum să folosiți instrumente pentru dezvoltarea software-ului în Linux și a descris cum să exploatați legarea dinamică în aplicațiile dumneavoastră. Acest capitol se oprește la alt subiect important de programare: cum se scriu aplicațiile cu interfață grafică de utilizator. Acest lucru înseamnă scrierea aplicațiilor X, deoarece X Window System este sistemul pe bază de ferestre esențial în Linux.

Interfața primară de programare a X Window System este biblioteca de funcții C Xlib. Puteți folosi funcții Xlib pentru a construi interfețe grafice de utilizator cu meniuri și butoane, folosind o ierarhie de ferestre; puteți afișa grafică, text și imagini în aceste ferestre. Cu toate că aveți multe posibilități cu Xlib, scrierea programelor folosind numai funcții Xlib poate fi plictisitoare. Pentru a genera interfețe cu utilizatorul fără să vă încurcați în detalii, aveți nevoie de funcții utilitare și de o colecție de componente predefinite, precum butoane, meniuri și bare de derulare. Din fericire, câteva pachete X furnizează instrumentele ce vă ajută să construiți ușor interfețe cu utilizatorul.

Versiunea actuală a setului de instrumente X conține pachetul Motif de la Open Software Foundation, XView de la Sun, OPEN LOOK Intrinsics Toolkit (OLIT) și Athena Widgets, pentru a aminti numai câteva dintre ele. Dintre acestea, Motif este cel mai popular pachet pentru crearea interfețelor grafice de utilizator.



Observație

Din păcate, Motif este un produs cu licență al Open Software Foundation, motiv pentru care CD-ROM-ul anexat acestei cărți nu cuprinde această aplicație. Dacă doriți să dezvoltați aplicații Motif, va trebui să cumpărați acest pachet separat. Cu toate acestea, vom discuta despre programarea Motif în acest capitol pentru că, dacă dezvoltați aplicații X, este probabil că va trebui să folosiți Motif.

Acest capitol oferă o scurtă descriere a setului de instrumente Motif, care este bazat pe biblioteca Xt Intrinsics. Acest capitol vă arată cum să creați și să manipulați primitive cu ajutorul funcțiilor furnizate de Xt Intrinsics. Cealaltă parte a capitolului oferă o imagine de

ansamblu a setului de primitive Motif și descrie cum să folosiți funcții Xlib pentru a desena într-o aplicație Motif.



Dacă nu ați cumpărat încă Motif pentru sistemul dumneavoastră Linux, capitolul 4 prezintă câțiva furnizori ai acestui produs.

Bazele programării Motif

Pachetul Motif furnizează un set de componente pentru interfața cu utilizatorul numite widgets (primitive). Puteți considera fiecare primitivă o fereastră cu câteva informații asociate, precum mărimea ferestrei și culoarea. Un strat de structuri de date și funcții construit cu Xlib, Xt Intrinsics, oferă structurile de date necesare pentru o primitivă. Xt Intrinsics execută automat redesenarea conținutului unei primitive atunci când este nevoie.

Exact cum ferestrele din X sunt organizate într-o ierarhie părinte-copil, cu ferestrele copil continute în cadrul limitelor ferestrei părinte, așa sunt organizate și primitivele Motif.

Dacă nu ați scris niciodată o aplicație cu interfață grafică de utilizator, va trebui să vă adaptați la un nou stil de programare atunci când scrieți aplicații Motif. Acest nou stil de programare este cunoscut ca *programare condusă prin evenimente*.



 Într-un program condus prin evenimente, nu specificați exact succesiunea sarcinilor pe care trebuie să le execute programul. În schimb, definiți multe funcții mici, fiecare din ele având rolul de a răspunde la un eveniment, precum „utilizatorul a executat clic pe butonul Save“. Programul condus prin evenimente așteaptă evenimentele și le procesează pe fiecare apelând subrutina corespunzătoare de tratare a evenimentelor.

Puteți înțelege cel mai bine programarea condusă prin evenimente parcurgând exemplul din secțiunea următoare.

Programarea Motif pas cu pas

Pentru a scrie un program Motif, parcurgeti următorii pași:

1. Includeti fisierele antet Motif, astfel:

```
# include <Xm/Xm.h>
```

```
/* Includeti fisierul header al fiecarei primitive pe care aveti de gand sa o
folositi.
```

```
* De exemplu, iata care sunt fisierele anet necesare
* pentru primitivile Form, Label si PushButton..
*/
```

```
# include <Xm/ Form.h>
# include <Xm/ Label.h>
# include <Xm/ PushB.h>
```

2. Apelați funcția `XtAppInitialize` pentru a inițializa setul de instrumente și creați o primitivă de nivel superior. Această funcție returnează un identificator (de tip `Widget`) care reprezintă primitiva de nivel superior.

3. Apelați `XtVaCreateManagedWidget` pentru a crea fiecare primitivă și setați variabilele interne ale acestora, cunoscute sub numele de *resurse*.

4. Apelați `XtAddCallback` pentru a indica ce funcții trebuie apelate pentru a trata anumite evenimente ce intervin într-o primitivă. Aceste funcții sunt cunoscute ca *funcții cu apel invers* (callback functions). Într-un program Motif toate operațiile sunt executate de funcțiile cu apel invers.

5. Apelați `XtRealizeWidget` cu identificatorul primitivei de nivel maxim ca argument. Această acțiune creează toate primitivele.

6. Apelați funcția `XtAppMainLoop` pentru a începe procesarea evenimentelor. Această acțiune începe rularea programului `Motif`.

Un program Motif simplu

Următorul listing arată un program Motif simplu ce afișează un buton. Programul se termină când utilizatorul execută clic pe buton.

```

/*-----*/
/* FiSier: xmquit.c
 *
 * Un program de exemplificare ce afiseaza un Buton de apasare(PushButton)
 * în interiorul unei primitive Form.
 *
 */

```

```
# include <Xm/Xm.h>
# include <Xm/ Form.h>
# include <Xm/ PushB.h>
```

```
/* Eticheta pentru butonul de apasare */
static char label[ ] = „Press here to quit.”;
```

```
/* Prototipul functiilor de apel invers */
static void cb_quit (Widget w, XtPointer client_data,
                    XtPointer call_data);
```

```
/*-----*/
void main (int argc, char **argv)
{
```

```

    XtAppContext app;
    Widget toplevel, quit_button, form;
    XmString xlabel;
/* Initializeaza kitul cu instrumente si creeaza primitiva de
    toplevel = XtAppInitialize (&app, „XMquit“, NULL, 0,
        &argc, argv, NULL, NULL, 0);

/* Creeaza primitiva Form*/
    form = XtCreateManagedWidget ("Form", xmFormWidgetClass,
        toplevel, NULL, 0);

```

```

/* Creeaza primitiva buton de apasare(push button) ca un copil al primitivei Form
* si-i seteaza eticheta in acelasi timp. Aceasta va arata, de asemenea,
* cum sa creati si sa folositi un „Motif String (XmString)“.
*/

```

```
xmlabel = XmStringCreateLtoR(label,  
    XmSTRING_DEFAULT_CHARSET),  
quit_button = XtVaCreateManagedWidget("Exit",  
    xmPushButtonWidgetClass, form,  
    xmNlabelString, xmLabel,
```

```

    NULL);
    XmStringFree (xmlabel);

/* Setează funcția de apel invers pentru acest buton de apăsare */
XtAddCallback (quit_button, XmNactivateCallback,
               cb_quit, NULL);

/* Realizează primitivă */
XtRealiseWidget (toplevel);

/* Porneste bucla */
XtAppMainLoop (app);
}
/* ----- */
/* c b _ q u i t
 *
 * Funcția ce va fi apelată când utilizatorul apasă
 * și eliberează butonul.
 */
static void cb_quit (Widget w, XtPointer client_data,
                    XtPointer call_data)
{
    XtCloseDisplay (XtDisplay (w));
    exit (0);
}

```

Fișierul Makefile pentru un program Motif

Folosiți fișierul makefile arătat în următorul listing pentru a compila și lega programul `xmquit`. (Este necesar ca Motif să fie instalat pe sistemul dumneavoastră)

```

#####
# Fișier Makefile pentru compilarea și legarea unui program Motif în Linux.

```

```
LDLAGS= -lXm -lXt -lX11 -lXext
```

```
all: xmquit
xmquit.o: xmquit.c
```

```
xmquit: xmquit.o
    cc -o xmquit xmquit.o $(LDLAGS)
```

Tastați `make` pentru a genera programul, astfel:

```
make
cc -c xmquit.c -o xmquit.o
cc o xmquit xmquit.o -lXm -lXt -lX11 -lXext
```

Pentru a testa programul, mai întâi asigurați-vă că X rulează pe PC-ul dumneavoastră Linux.

Dintr-o fereastră `xterm`, tastați `xmquit` pentru a lansa în execuție programul. Figura 23-1 prezintă interfața grafică afișată de programul `xmquit`.

Când executați clic pe buton programul se termină.

Resursele unei primitive

Când folosiți o primitivă Motif într-o aplicație, nu este necesar să cunoașteți detalii despre structura de date a primitivei. Ceea ce vă trebuie sunt informațiile despre parametrii configurabili ai acesteia. Acești parametri, cunoscuți ca resurse ale primitivei, controlează

aspectul și comportamentul acesteia. Resursele unei primitive înseamnă, pur și simplu, orice dată configurabilă folosită de aceasta.

Figura 23-1

Rezultatul programului Motif demonstrativ `xmquit`.



Pentru a fi folosite ca element constructiv al interfețelor cu utilizatorul, o primitivă trebuie să fie extrem de configurabilă. O primitivă care permite programatorului să aleagă culorile pentru prim-plan (foreground) și fundal (background), de exemplu, este mult mai folosită decât una pentru care aceste valori nu pot fi modificate.

Ca programator, puteți seta resursele unei primitive prin intermediul unei liste de argumente. Pentru aceasta, trebuie să consultați mai întâi documentația primitivei și să învățați numele resurselor pe care doriți să le setați. Numele resurselor sunt constante, precum `XmNwidth` și `XmNheight`. Folosiți constantele `XmNwidth` și `XmNheight` pentru a vă referi la resursele de lățime și înălțime ale unei primitive.

Pentru a seta valorile resurselor printr-o listă de argumente, specificați valoarea fiecărei resurse într-o structură `Arg`, definită în fișierul `<X11/Intrinsic.h>` cu următoarea declarație `typedef`:

```

typedef struct
{
    String name; /* Numele resursei */
    XtArgVal value; /* Valoarea ei */
} Arg, *ArgList;

```

Valoarea resursei este stocată ca `XtArgVal` – un tip de date dependente de sistem, capabile de a stoca un pointer către orice variabilă C. Dacă valoarea unei resurse este mai mică decât mărimea `XtArgVal`, atunci ea este stocată direct în câmpul `value` al structurii `Arg`. În caz contrar, câmpul `value` este un pointer al valorii resursei.



Observație

■ Folosiți un șir de structuri `Arg` inițializat static.

■ Atribuiți valorile în timpul rulării, folosind macro-ul `XtSetArg`.

Pentru a vedea cum se setează resursele, considerați primitivă `PushButton`. Să presupunem că doriți să setați lățimea și înălțimea butonului de presiune, precum și *eticheta* (label) acestuia – textul afișat pe buton. Documentația primitivei `PushButton` vă informează că numele resurselor sunt `XmNwidth`, `XmNheight` și `XmNlabelString`. Trebuie să știți, de asemenea, că șirul pentru `XmNlabelString` nu este un simplu șir de caractere C; este un *șir compus* (compound string) – un tip special de date creat prin transmiterea șirului ca argument unei rutine Motif (`XmStringCreateLtoR`).

Având această informație, puteți seta valorile resurselor cu un masiv `Arg` astfel:

```

Arg args[20];
Cardinal nargs;
XmString cstr;

...

cstr = XmStringCreateLtoR ("WarnGen",
                          XSTRING_DEFAULT_CHARSET);
XtSetArg (args [nargs], XmNwidth, 160); nargs++;
XtSetArg (args [nargs], XmNheight, 80); nargs++;

```

```
XtSetArg (args [nargs], XmNlabelString, cstr); nargs++;
XmStringFree (cstr);
```



Există un motiv pentru care `nargs++` nu este folosit în `XtSetArg` pentru a incrementa contorul argumentelor; `XtSetArg` este definit ca un macro, astfel încât folosește primul argument de două ori. Dacă folosiți `nargs++` în primul argument, macro-ul va incrementa `nargs` de două ori la fiecare apel. De aceea, cele mai multe aplicații definesc lista de argumente așa cum s-a arătat anterior.

După ce pregătiți lista de argumente, apăsați funcția `XtCreateManagedWidget` pentru a crea o primitivă și a-i seta resursele, astfel:

```
Arg args [20]
Cardinal nargs;
Widget toplevel, /*Primitiva de nivel superior creata anterior */
pbl; /* Un buton de apasare nou */

pbl = XtCreateManagedWidget ("WarnGen", xmPushButtonWidgetClass,
toplevel,
args, nargs);
```

Această procedură creează un nou buton de presiune denumit `WarnGen` a cărui primitivă părinte este de nivel superior și ale cărui setări ale resurselor inițiale sunt în șirul `args`.



Puteți evita etapele plictisitoare de specificare a resurselor una câte una, urmate de crearea primitivelor. Biblioteca `XtIntrinsics` conține funcția `XtVaCreateManagedWidget`, care vă permite să specificați resursele și să creați primitiva cu un singur apel de funcție. Puteți crea un buton de presiune cu lățime, înălțime și etichetă specificate, de exemplu, folosind următorul apel al funcției `XtVaCreateManagedWidget`:

```
Widget toplevel, /* Primitiva de nivel superior creata anterior */
pbl; /* Buton de apasare nou */
XmString label; /* Sir compus pentru stocarea etichetei */

/* Mai intai, creati sirul compus pentru eticheta */
xmlabel = XmStringCreateLtoR ("Wargen", XmSTRING_DEFAULT_CHARSET),

h_button = XtVaCreateManagedWidget ("WarnGen",
xmPushButtonWidgetClass, toplevel,
XmNwidth, 200,
XmNheight, 100,
XmNlabelString, xmlabel,
NULL);

/* Eliberati sirul compus (kitul cu instrumente Motif face o copie a acestuia) */
XmStringFree (xmlabel);
```

Așa cum o arată fragmentul de cod de mai sus, `XtVaCreateManagedWidget` acceptă un număr variabil de argumente astfel:

- Se solicită primele trei argumente, similare celor cerute de `XtCreateManagedWidget`.
- Primul argument este numele primitivei, al doilea este clasa primitivei, iar al treilea specifică părintele primitivei.
- După cele trei argumente obligatorii se găsește o listă de specificații a resurselor. Fiecare specificație este sub forma unui nume de resursă, urmat de valoarea acelei resurse.
- Un nume de resursă `NULL` marchează sfârșitul listei.

Înregistrarea apelurilor inverse

Cele mai multe primitive Motif conțin o clasă de resurse cunoscute ca apeluri inverse (callbacks), care sunt pointeri către funcții. Puteți seta o astfel de resursă pe una din funcțiile dumneavoastră, iar primitiva apelează această funcție ca răspuns la unul sau mai multe evenimente. Aceste funcții sunt cunoscute sub numele de *funcții de apel invers* (callback functions), deoarece setul de instrumente Motif le apelează invers când este nevoie.

O primitivă are, de obicei, mai multe tipuri de resurse de apel invers, fiecare tip fiind destinat pentru apelarea funcțiilor într-o anumită situație. De fapt, resursele de apel invers ale unei primitive reprezintă o listă de funcții, nu o singură funcție.

Primitiva apelează toate apelurile inverse, când condițiile pentru acele resurse de apel invers sunt îndeplinite. Ordinea de apelare este aceeași cu ordinea în care înregistrați apelurile inverse.

În programul prezentat în secțiunea „Un program Motif simplu”, funcția `cb_quit` este o funcție de apel invers pentru resursa primitivă `XmNactivateCallback` a primitivei butonului de apăsare. În conformitate cu documentația `PushButton`, primitiva apelează funcțiile din resursa `XmNactivateCallback` atunci când utilizatorul execută clic pe buton. Pe lângă `XmNactivateCallback`, `PushButton` mai are încă două resurse de apel invers:

- Funcțiile `XmNarmCallback`, apelate când utilizatorul execută clic pe buton.
- Lista `XmNdisarmCallback`, apelată când utilizatorul eliberează butonul.

Așa cum se arată în exemplu, folosiți funcția `XtAddCallback` pentru a adăuga o funcție la o listă de apeluri inverse ale unei primitive. Pentru a adăuga funcția `cb_quit` la resursa `XmNactivateCallback` a primitivei `quit_button`, folosiți următoarele:

```
XtAddCallback (quit_button, XmNactivateCallback, cb_quit, NULL);
```

Ultimul argument al funcției `XtAddCallback` – declarat de tipul `XtPointer` – este un pointer către datele care se vor transmite funcției de apel invers când primitiva o apelează. Funcția de apel invers, `cb_quit`, are următorul prototip:

```
static void cb_quit (Widget w, XtPointer client_data, XtPointer
call_data);
```

Când primitiva apelează această funcție, al doilea argument va reprezenta datele transmise către `XtAddCallback` ca ultim argument. Ultimul argument transmis funcției de apel invers este pointerul către o structură `XmAnyCallbackStruct`, care este definită în `<Xm/Xm.h>` (se referă la fișierul `/usr/include/Xm/Xm.h`) astfel:

```
typedef struct
{
int reason; /* Indica de ce s-a apelat apelul invers */
XEvent event; /* Informatii despre evenimentul care a generat apelul invers */
} XmAnyCallbackStruct;
```



Observație

Câmpul `reason` indică de ce primitiva a apelat funcția de apel invers. Trebuie să consultați documentația primitivei pentru a interpreta valoarea acestui câmp. Câmpul `event` este un pointer către o structură `Xevent` cu informații despre evenimentul care a generat apelul invers.

Înregistrarea subrutinei de tratare a evenimentelor

Să presupunem că permiteți utilizatorului să deseneze în fereastra primitivei cu mouse-ul. (Motif funziona o primitivă `XmDrawingArea` în acest scop.). Pentru a programa această facilitare, doriți să capturați efectuarea unui clic de mouse în fereastra primitivei.

Setul de instrumente Motif vă permite aceasta folosind o metodă asemănătoare cu resursele de apel invers folosite de primitive. De fapt, vă puteți înregistra propria subrutină de tratare a evenimentelor pentru evenimentele selectate în fereastra primitivei. Apoi, când aceste evenimente au loc, primitiva apelează subrutina înregistrată de tratare a evenimentelor, dându-vă șansa să intrați în acțiune.

Folosiți funcția `XtAddEventHandler` din biblioteca `XtIntrinsics` pentru a adăuga o subrutină de tratare a evenimentelor. Pentru a adăuga o funcție numită `event_handler` ca subrutină pentru tratarea evenimentelor `ButtonPress` din primitiva `drawing`, veți folosi următoarele:

```
XtAddEventHandler (drawing, ButtonPressMask, FALSE,
    event_handler, NULL);
```

Primul argument al funcției `XtAddEventHandler` este primitiva pentru care creați o subrutină de tratare a evenimentelor. Al doilea argument este o *mască a evenimentului* (un model de biți reprezentat de o valoare constantă denumită) care specifică evenimentele pentru care este invocată subrutina de tratare a acestora. Al treilea argument este o valoare booleană care ar trebui setată pe valoarea `True` dacă setați subrutina de tratare a evenimentelor pentru unul din evenimentele `X_ClientMessage`, `MappingNotify`, `SelectionClear` sau `SelectionRequest` – pentru care nu există nici o mască a evenimentului. În acest caz, setați al doilea argument pe valoarea `NoEventMask`. Ca și la funcția `XtAddCallback`, ultimul argument este un pointer către toate datele pe care le doriți transmise înapoi către subrutina de tratare a evenimentelor, atunci când aceasta este apelată.



Trebuie să scrieți funcția `event_handler` în conformitate cu următorul prototip:

```
void event_handler (Widget w, XtPointer client_data, XEvent *event);
```

Observație

Așa cum arată această funcție prototip, primitiva apelează subrutina de tratare a evenimentelor cu trei argumente:

- Primul argument este identificatorul primitivei.
- Al doilea argument este același pointer pe care l-ați transmis ca ultim argument al funcției `XtAddEventHandler` când ați înregistrat această subrutină de tratare a evenimentelor.
- Al treilea argument este un pointer către structura `XEvent` care a generat apelul funcției.

Veți găsi un rezumat al evenimentelor X în secțiunea „Rezumatul evenimentelor X”.

Primitive Motif

Un set de primitive este ca orice cutie cu instrumente; înainte de a folosi instrumentele, trebuie să știți care este rolul fiecăruia. Această secțiune vă furnizează o imagine generală despre primitivele Motif astfel încât să puteți cunoaște categoriile de primitive și rolul fiecăreia. De asemenea, trebuie să învățați să folosiți primitivele importante din fiecare categorie. Apoi veți fi capabil să alegeți primitivele care îndeplinesc cerințele aplicațiilor dumneavoastră. Următoarele secțiuni vă oferă o imagine generală asupra primitivelor Motif.

Există trei categorii distincte de primitive Motif:

- *Primitivele Shell* furnizează fereastra de nivel superior pentru o aplicație Motif. Casetele de dialog pop-up folosesc, de asemenea, primitive `shell`.
- *Primitivele autonome* reprezintă primitive de sine stătătoare, precum etichete, butoane de presiune și bare de derulare.

- *Primitivele manager* pot conține alte primitive copil și gestionează amplasarea primitivelor copil. Această categorie include formulare, casete de mesaje și ferestre derulante.



Observație

Toate primitivele Motif moștenesc resursele unei primitive denumite `Core`. Cunoașterea resurselor primitivei `Core` este importantă, deoarece fiecare primitivă dispune de aceste resurse. În continuare este prezentată o listă a resurselor `Core`:

- `XmNaccelerators` este tabela de conversie care leagă o succesiune de evenimente ale tastaturii și mouse-ului cu acțiuni specifice.
- `XmNancestorSensitive` este o variabilă booleană care indică dacă părintele imediat al unei primitive primește evenimente de intrare. Pentru a schimba setarea acestei resurse, apăsați funcția `XtSetSensitive`.
- `XmNbackground` este valoarea pixelului din fundal (`background`) pentru fereastra primitivei. Valoarea pixelului este convertită, prin intermediul hărții de culori curente, într-o culoare RGB efectivă.
- `XmNbackgroundPixmap` este o hartă de pixeli (un bloc de memorie care stochează un șir cu valori ale pixelilor) folosită pentru a acoperi fereastra primitivei.
- `XmNborderColor` și `XmNborderPixmap` reprezintă harta de culori și harta de pixeli pentru marginea ferestrei primitivei.
- `XmNborderWidth` este lățimea marginii ferestrei primitivei.
- `XmNcolormap` este harta de culori folosită de fereastra primitivei.
- `XmNdepth` reprezintă numărul de biți folosit pentru fiecare valoare de pixel din fereastra primitivei. Această valoare este setată de `XtIntrinsics` atunci când este creată fereastra.
- `XmNdestroyCallback` este lista funcțiilor ce trebuie apelate când distrugeți primitiva prin apelarea funcției `XtDestroyWidget`.
- `XmNheight` și `XmNwidth` reprezintă înălțimea și lățimea ferestrei primitivei, exclusiv grosimea marginii.
- `XmNmappedWhenManaged` este un indicator care, când este setat pe valoarea `True`, afișează fereastra primitivei imediat ce aceasta este realizată și gestionată. (Funcția `XtCreateManagedWidget` gestionează primitiva.) Puteți modifica acest indicator apelând funcția `XtSetMappedWhenManaged`.
- `XmNscreen` este un pointer către structura de date `Screen`, ce conține informații despre ecranul fizic în care este afișată fereastra primitivei.
- `XmNsensitive` este o variabilă booleană care, atunci când are valoarea `True`, determină `XtIntrinsics` să trimită evenimente ale mouse-ului și tastaturii către primitivă. Pentru a modifica această resursă, folosiți funcția `XtSetSensitive`.
- `XmNtranslations` este *tabela de conversie* – o listă de evenimente cu funcțiile corespunzătoare apelate atunci când evenimentul are loc.
- `XmNx` și `XmNy` sunt coordonatele `x` și `y` ale colțului din stânga-sus al ferestrei primitivei, exclusiv marginea. Coordonatele sunt specificate în cadrul cu coordonate părinte ale primitivei.

Primitive shell

O primitivă `shell` administrează numai un copil; scopul ei principal este de a crea fereastra de nivel superior a unei aplicații. `XtIntrinsics` furnizează câteva clase de primitive `shell`, cele mai importante fiind următoarele:

- TopLevelShell
- OverrideShell
- TransientShell

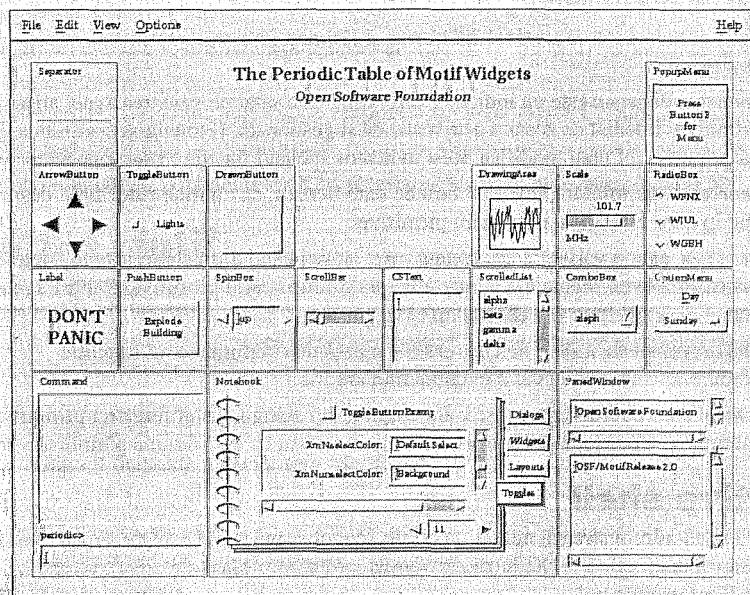
Când apelați funcția `XtAppInitialize`, `Xt` `Intrinsics` creează un exemplu de primitivă `ApplicationShell`, o subclasă a clasei de primitive `TopLevelShell`. Primitivile `ApplicationShell` și `TopLevelShell` sunt folosite pentru ferestrele de nivel superior normale ale aplicațiilor. Gestionarul de ferestre interacționează cu aceste ferestre de nivel superior.

Clasa `TransientShell` este folosită pentru casetele de dialog pop-up, care sunt ferestre de nivel superior pe care utilizatorul le poate deplasa, dar nu le poate reduce la mărimea unei pictograme. Pentru a crea o casetă de dialog, folosiți o subclasă a clasei `TransientShell` numită `DialogShell`. `Motif` include câteva funcții utile, care vă permit să creați o primitivă `DialogShell` și să o plasați în interiorul altei primitive, precum o casetă de selecție sau o casetă de mesaje. Aceste funcții utile au nume care încep cu `XmCreate` și se termină cu `Dialog` (de exemplu, `XmCreateFileSelectionDialog` sau `XmCreateMessageDialog`).

Versiuni de Motif

Cea mai recentă versiune a programului `Motif` este 2.0. Totuși, o mare parte din utilizatorii UNIX folosesc încă versiunea 1.2.x. Un motiv pentru persistența versiunii mai vechi a aplicației `Motif` este `Common Desktop Environment (CDE)` care folosește `X11R5` și `Motif 1.2` ca fundament. Puteți obține `Motif 2.0` atunci când cumpărați o versiune comercială pentru Linux.

Majoritatea distribuțiilor `Motif` includ un program de exemplificare numit periodic care afișează un tabel periodic al primitivelor (mai știți sistemul periodic al elementelor de la chimie?). În figura de mai jos este prezentată o captură de ecran ce conține tabelul periodic de primitive pentru `Motif 2.0`.



În comparație cu versiunea 1.2.x, `Motif 2.0` are următoarele caracteristici noi:

- Adăugarea a șase primitive noi: `CSText` (editare de text compusă), `ComboBox` (exact ca și `ComboBox` din `Microsoft Windows`, o listă derulantă cu o zonă de editare a textului), `spinbutton` (două butoane cu săgeți pentru modificarea opțiunilor), `IconGadget` (browser cu listă bazat pe pictograme și text), `container` (administrează primitivile `IconGadget` și `Notebook` (afișează informații într-o carte cu pagini cu tabulatoare)).
- Suport pentru formatul imaginilor `X Pixmap (XPM)` pentru a defini culorile pictogramelor.

- Suport unificat pentru tăiere și lipire (cut-and-paste) precum și pentru tragere-si-plasare (drag-and-drop), prin intermediul modelului `Uniform Transfer Model (UTM)`.
- Suport de la tastatură pentru operația de tragere-si-plasare (drag-and-drop).
- Folosirea proprietății `XmRenderTable` pentru a putea gestiona fonturi multiple, culori, sublinieri și marcaje de tabulare în tipul de date `XmString`.
- Includerea claselor C++ `XmCxxPrimitive` și `XmCxxManager`, din care puteți deriva alte clase de primitive C++.

Primitiva `OverrideShell` este un tip de primitivă shell care ignoră complet administratorul de ferestre. Administratorul de ferestre nu inserează nici un cadru în jurul primitivei `OverrideShell`. `Motif` definește o subclasă a clasei de primitive `OverrideShell-XmMenuShell`, pentru a afișa meniurile pop-up. Puteți folosi funcția `XmCreatePopupMenu` pentru a crea un `MenuShell` cu un meniu inclus în el.

Primitive autonome

Toate primitivile autonome `Motif` sunt derivate din clasa `XmPrimitive`, care la rândul ei este definită ca o subclasă a clasei `Core`. Clasa `XmPrimitive` definește un set de resurse standard moștenite de toate primitivile autonome. În particular, clasa `XmPrimitive` este responsabilă pentru efectul de umbră 3D (3-D shading) ale primitivelor `Motif`.



Observație

Următoarele opt primitive autonome sunt derivate direct din clasa `XmPrimitive`:

- `ArrowButton` afișează un buton cu o săgeată. Puteți specifica direcția săgeții.
- `CSText` afișează și modifică linii multiple ale șirurilor compuse. Textul poate include fonturi multiple, culori, text subliniat și tabulatoare înglobați.
- `Label` afișează un șir sau o hartă de pixeli în interiorul unei ferestre.
- `List` afișează o listă din care utilizatorul poate selecta unul sau mai multe articole text.
- `Sash` este folosit ca separator între două ferestre. Utilizatorul poate trage separatorul pentru a ajusta mărimea ferestrelor.
- `ScrollBar` este o bară de derulare cu un glisor și săgeți la cele două capete. Barele de derulare pot fi orizontale sau verticale.
- `Separator` este folosit pentru separarea articolelor dintr-un meniu.
- `Text` acționează ca un editor pentru o singură linie sau linii multiple.
- `TextField` afișează o singură linie de text ce poate fi editat.

Funcționalitatea clasei `Label` este detaliată în continuare de următoarele subclase:

- `CascadeButton` afișează un meniu derulant asociat când utilizatorul execută clic pe `CascadeButton`.
- `DrawnButton` este un buton ce vă permite să desenați pe el (astfel încât puteți afișa pe el orice doriți).
- `PushButton` este un buton de presiune standard.
- `TearOffButton` este un buton de presiune special, care afișează un meniu detașabil.
- `ToggleButton` afișează un șir sau o hartă de pixeli lângă un buton mic. Butonul reprezintă un comutator cu două valori: 0 și 1. Când utilizatorul execută clic pe buton, el schimbă de pe 1 pe 0 sau invers.

Toate primitivile autonome moștenesc resursele clasei `XmPrimitive`. Iată în continuare câteva resurse importante ale clasei `XmPrimitive`:

- `XmNbottomShadowColor` este valoarea pixelului folosită pentru a desena marginea de jos și din dreapta a umbrei unei primitive autonome. Aceste două margini sunt numite `bottom shadow` (umbra părții inferioare).
- `XmNbottomShadowPixmap` este harta de pixeli folosită pentru a desena umbra din partea inferioară.
- `XmNforeground` este culoarea pentru prim-plan folosită în primitivile autonome.
- `XmNhelpCallback` este lista apelurilor inverse care pot fi legate de funcții ce furnizează asistență contextuală.
- `XmNhighlightColor` este culoarea folosită pentru a evidenția fereastra primitivei. Un buton de presiune, de exemplu, este evidențiat atunci când utilizatorul execută clic pe el.
- `XmNhighlightOnEnter` este o variabilă booleană care, atunci când este setată pe valoarea `True`, cere primitivei să-și evidențieze fereastra când cursorul intră în interiorul ei (cu condiția ca focalizarea de intrare să urmeze indicatorul; fereastra cu focalizarea de intrare prin intrarea de la tastatură.) Această resursă este ignorată dacă utilizatorul trebuie să execute clic pentru a indica fereastra activă.
- `XmNhighlightPixmap` este harta de pixeli folosită pentru a evidenția fereastra primitivei.
- `XmNhighlightThickness` este grosimea dreptunghiului folosit pentru evidențierea ferestrei primitivei.
- `XmNnavigationType` este o constantă care specifică atribuirea focalizării către primitivă în timpul traversării tastaturii (traversarea tastaturii se referă la modul în care focus-ul se deplasează de la o primitivă la alta pe măsură ce utilizatorul apasă tasta `Tab`); această constantă poate fi `XmNONE`, `XmTAB_GROUP`, `XmSTICKY_TAB_GROUP` sau `XmEXCLUSIVE_TAB_GROUP`.
- `XmNshadowThickness` este grosimea umbrei.
- `XmNtopShadowColor` este culoarea pentru umbra din partea superioară. Este folosită numai dacă `XmNtopShadowPixmap` este `NULL`.
- `XmNtopShadowPixmap` este harta de pixeli pentru umbra din partea superioară.
- `XmNtraversalOn` este o valoare booleană ce indică dacă primitiva acceptă intrări de la tastatură.
- `XmNunitType` este unitatea de măsură folosită pentru toate valorile ce specifică dimensiuni în primitivă. Valoarea prestabilită este copiată de la un administrator care este proprietarul primitivei. De obicei, unitatea prestabilită este pixelul (specificată prin constanta `XmPIXELS`). Motif vă permite să lucrați cu unități independente de dispozitiv, dacă preferați acest lucru. Specificați unitatea cu una din constantele: `XmPIXELS`, `Xm100TH_MILLIMETERS`,

`Xm1000TH_INCHES`, `Xm100TH_POINTS` sau `Xm100TH_FONT_UNIT`, definite în fișierul antet `<Xm/Xm.h>`. Puteți seta resursa `XmNunitType` pe `Xm100TH_FONT_UNIT`, de exemplu, pentru a indica faptul că toate dimensiunile sunt definite în sutimi de unități utilizate pentru mărimea fontului. Puteți seta explicit unitatea de măsură pentru mărimea fontului cu funcția `XmSetFontUnits`.

- `XmUserData` este un pointer ce nu este folosit intern. În această resursă puteți stoca un pointer către propriile dumneavoastră date specifice primitivei.

Primitive manager

Primitivile manager sunt derivate din clasa `XmManager` – o subclasă a clasei `Constraint` care, la rândul ei, provine din clasa `Composite`. O primitivă manager poate acționa ca un container pentru alte primitive copil și poate administra aranjarea acestora. Câteva primitive manager, precum `Form`, vă permit să specificați aranjarea în ceea ce privește limitele, ca de exemplu „atașați această primitivă copil la marginea din stânga a containerului.”

Resursele tuturor primitivelor manager sunt moștenite din clasa `XmManager`. Resursele `XmManager` sunt asemănătoare cu cele din clasa `XmPrimitive`.

În continuare, sunt prezentate câteva primitive manager importante:

- Primitiva `Command` este un fel de casetă de selecție care furnizează un mecanism pentru istoria comenzilor. O zonă este destinată introducerii comenzilor de către utilizatori. Când executați o comandă apăsând `Enter`, aceasta este salvată într-un buffer de istorie, care este afișat într-o listă derulantă. Puteți folosi primitiva `Command` pentru a accepta datele introduse de utilizator pentru aplicațiile dumneavoastră conduse prin comenzi.
- Primitiva `DrawingArea` furnizează o fereastră vidă în care puteți desena grafică sau text (folosind funcțiile `Xlib` descrise în secțiunea „Generalități despre funcțiile `Xlib`” din acest capitol).
- Primitiva `FileSelectionBox` este un tip special de casetă de selecție, care afișează o listă cu nume de fișiere. Primitiva are o zonă în care utilizatorul poate introduce un filtru pentru fișier, care reprezintă un șir folosit pentru căutarea fișierelor pe care dorim să le localizăm (.c, de exemplu, afișează toate numele de fișiere care se termină cu .c). Selecția curentă a utilizatorului este afișată în altă casetă. În partea inferioară a primitivei sunt patru butoane cu etichetele `OK`, `Filter`, `Cancel` și `Help`.
- Primitiva `Form` suportă aranjări complexe specificate prin anumite resurse pe care primitiva le atașează fiecăruia din copii ei. Aceste resurse sunt cunoscute ca resurse de limitare; ele furnizează un limbaj de aranjare pe care-l puteți folosi pentru a indica amplasarea primitivelor copil în primitiva `Form`. Fiecare resursă de limitare indică relația spațială a unei primitive copil cu primitiva `Form` sau cu altă primitivă copil existentă.
- Primitiva `Frame` afișează un cadru (cu umbră 3D) în jurul unei singure primitive copil.
- Primitiva `MainWindow` furnizează o aranjare standard pentru fereastra principală a unei aplicații. Primitiva `MainWindow` gestionează o combinație dintre o bară de meniuri, o primitivă `Command`, o primitivă `DrawingArea` și bare de derulare.
- Primitiva `MessageBox` afișează un mesaj și un rând cu trei butoane cu etichetele `OK`, `Cancel` și `Help`.
- Primitiva `PanedWindow` își aranjează copiii în ferestre dispuse vertical.
- Primitiva `RowColumn` își aranjează copiii în rânduri și coloane. Puteți folosi primitivile `RowColumn` în bare de meniuri și meniuri derulante.

- Primitiva `Scale` afișează un dreptunghi alungit, cu un glisor ce permite utilizatorului să introducă o valoare numerică. Puteți seta valoarea minimă și maximă pentru scară și pentru valorile în virgulă mobilă și puteți specifica numărul de zecimale.
- Primitiva `Scrolled Window` gestionează o suprafață de lucru (care poate conține alte primitive) și două bare de derulare (orizontală și verticală).
- Primitiva `SelectionBox` afișează o listă de articole într-o casetă derulantă și furnizează o suprafață în care este afișată selecția curentă. Primitiva are trei butoane cu etichetele `OK`, `Cancel` și `Help`. Opțional, puteți activa sau dezactiva un al patrulea buton cu eticheta `Apply`.

Xlib și Motif

Puteți folosi primitivele Motif pentru a construi interfața cu utilizatorul a unei aplicații Motif, dar majoritatea aplicațiilor autentice nu pot fi construite numai cu primitive Motif. Să presupunem că aplicația dumneavoastră este destinată vizualizării imaginilor din satelit pentru prognoza meteo. Puteți folosi primitive Motif pentru a permite utilizatorului să selecteze o imagine din satelit și chiar să pregătească o suprafață de vizualizare derulantă pentru a afișa imaginea. Totuși, pentru a afișa imaginea, trebuie să vă bazați pe funcțiile Xlib. Puteți folosi aceste funcții Xlib pentru a afișa imagini, grafică și text în aplicații Motif. În plus, este necesar să scrieți instrucțiuni de cod ce răspund unui eveniment X.



Observație

Trebuie să învățați bazele programării X cu funcții Xlib, chiar dacă folosiți setul de instrumente Motif pentru a scrie majoritatea aplicațiilor dumneavoastră. Următoarele secțiuni vă oferă o imagine generală despre Xlib și evenimentele X. De asemenea, veți vedea un exemplu de utilizare a funcțiilor de desenare Xlib într-o aplicație Motif.

O imagine generală despre Xlib

Sistemul X Window System folosește un model client/server pentru a-și oferi serviciile. Serverul X se ocupă de introducerea și afișarea datelor pe ecran. Aplicațiile X reprezintă clienții care folosesc facilitățile serverului trimițând cereri cu protocolul X. Aceste cereri sunt livrate printr-o legătură de comunicație între clienți și serverul X. Dacă serverul și clientul rulează pe calculatoare diferite, această legătură este o conexiune de rețea. Dacă serverul și clientul se găsesc pe același calculator, comunicația se poate face printr-un bloc de memorie partajat sau printr-un mecanism de comunicare între procese (IPC) ce poate fi utilizat de sistemul de operare.

Cu toate că este posibil să scrieți o aplicație X care realizează toate intrările și ieșirile trimițând cereri de protocol X către server, această operație este sinonimă cu programarea unui microprocesor în limbajul lui mașină. Pentru a simplifica munca programatorului, sistemul X Window System include biblioteca Xlib, un set de funcții și macro-uri ale limbajului C pe care aplicațiile X le pot folosi pentru a accesa facilitățile serverului X.

Deoarece Motif se bazează pe X, o aplicație Motif este de asemenea un client X; indirect, o aplicație Motif apelează funcțiile Xlib.

Scopul principal al bibliotecii Xlib este de a oferi o modalitate facilă pentru programatorii C de a trimite cereri de protocol X către server. Totuși, biblioteca Xlib este mult mai mult decât un set de funcții cu o corespondență unu-la-unu cu toate cererile posibile de protocol X; această bibliotecă include, de asemenea, multe funcții utile pentru a facilita realizarea sarcinilor uzuale, în timp ce protocolul X este ascuns complet de programator. De exemplu, există o singură cerere de protocol X pentru a crea o fereastră, dar Xlib are două rutine pentru crearea

ferestrelor: `XCreateSimpleWindow` și `XCreateWindow`, prima mai simplă decât a doua. Asemănător, culorile pentru prim-plan și fundal ale unei ferestre sunt specificate într-un context grafic. Pentru a seta aceste culori, pur și simplu apelați rutinele `Xlib` `XSetForeground` și `XSetBackground`. Xlib are grijă de setarea cererilor de protocol adecvate pentru a schimba numai aceste culori în contextul grafic specificat.

Xlib include, de asemenea, multe funcții utilitare care nu au nimic de-a face cu interacțiunea cu serverul X. Funcțiile utilitare Xlib vă ajută să obțineți preferințele utilizatorilor dintr-un fișier resursă, să manipulați imaginile de pe ecran și hărțile de biți (bitmaps), să transformați numele culorilor în valori ale pixelilor și să utilizați administratorul de resurse.

Informații suplimentare despre Xlib și programarea Motif

Acest capitol vă oferă o imagine de ansamblu despre programarea Motif și modul de folosire a funcțiilor Xlib în aplicațiile Motif. Totuși, un singur capitol este insuficient pentru a vă oferi toate informațiile de care aveți nevoie pentru a exploata la maximum puterea Xlib și setul de instrumente Motif. Datorită popularității Motif și X, există foarte multe cărți despre Xlib și programarea Motif. Pentru informații mai detaliate despre Xlib și setul de instrumente Motif, consultați oricare dintre următoarele cărți:

- Volume 2: *Xlib Reference Manual (Manual de referință Xlib)*, O'Reilly & Associates, 1993

- Volume 6A: *Motif Programming Manual (Manual de programare Motif)*, Second Edition, Dan Heller, Paula Ferguson și David Brennan, O'Reilly & Associates, 1994

- Volume 6B: *Motif Reference Manual (Manual de referință Motif)*, Paula Ferguson și David Brennan, O'Reilly & Associates, 1993

- Volume 1: *Xlib Programming Manual (Manual de programare Xlib)*, O'Reilly & Associates, 1993

Generalități despre funcțiile Xlib

În esență, funcțiile Xlib permit unei aplicații să deschisă afișarea X (să se conecteze la serverul X), să creeze ferestre și să deseneze în ele, să regăsească evenimente și în cele din urmă să oprească afișarea. Dacă dezvoltăți o aplicație Motif, vă puteți descurca cu un număr destul de mic de funcții Xlib. Nu trebuie să vă faceți probleme cu pornirea afișării, de exemplu, dacă folosiți setul de instrumente Motif. Într-o aplicație Motif, folosiți în primul rând funcțiile de grafică ale bibliotecii Xlib, imaginile și funcțiile al căror rezultat este sub formă de text.

Tabelul 23-1 rezumă cele mai folosite funcții Xlib, grupate în funcție de sarcina pe care o îndeplinesc. Această listă nu este completă, deoarece este cam lungă. Ca programator Motif nu trebuie să învățați toate aceste funcții; eu vă pun la dispoziție această listă doar pentru a vă face o imagine de ansamblu despre facilitățile bibliotecii Xlib. Pentru o listă completă a funcțiilor Xlib, consultați cărțile prezentate în secțiunea „Informații suplimentare despre Xlib și programarea Motif”.

Tabelul 23-1 Funcții Xlib uzuale, grupate în funcție de operația efectuată

Operație	Funcții Xlib uzuale
Pornește/oprește afișarea X	<code>XOpenDisplay</code> <code>XCloseDisplay</code>
Preia preferințele utilizatorului	<code>XGetDefault</code> <code>XrmInitialize</code> <code>XrmGetResource</code> <code>XrmParseCommand</code>

(continuare)

Tabelul 23-1 Continuare

Operație	Funcții Xlib uzuale
Creează/gestionează ferestre	XCreateSimpleWindow XCreateWindow XDestroyWindow XMapRaised XMapWindow XUnmapWindow XGetWindowAttributes XChangeWindowAttributes
Controlează mărimea și poziția ferestrei	XGeometry XGetGeometry XMoveWindow XResizeWindow XMoveResizeWindow XRaiseWindow XCirculateSubWindows
Interacționează cu programul de gestiune a ferestrelor	XSetStandardProperties XGetWMHints XSetWMHints
Preia evenimente și le procesează	XSelectInput XEventsQueued XNextEvent XPollEvent
Se sincronizează cu serverul	XFlush XSync XSynchronize
Tratează erorile	XGetErrorText XSetErrorHandler XSetIOErrorHandler
Manipulează contexte grafice	XCreateGC XChangeGC XCopyGC XFreeGC XSetForeground XSetState XSetBackground XSetFunction
Desenează elemente grafice	XDrawArc XDrawArcs XDrawLine XDrawLines XSetDashes XSetArcMode XDrawPoints XDrawPoint XDrawRectangle XDrawRectangles XFillArc XFillArcs XSetFillRule XFillRectangle XFillRectangles XFillPolygon XSetLineAttributes
Șterge și copiază anumite suprafețe	XClearWindow XClearArea XCopyArea XCopyPlane
Desenează text	XLoadFont XLoadQueryFont XSetFont XUnloadFont XDrawString XDrawImageString XDrawText XTextWidth XListFonts
Folosește culori	XDefaultColorMap XDefaultVisual XGetVisualInfo XParseColor XAllocColor XGetStandardColorMap XSetWindowColorMap XFreeColors
Afișează imagini	XCreatePixmap XFreePixmap XCreatePixmapFromBitmapData XReadBitmapFile XWriteBitmapFile XCreateImage XDestroyImage XGetImage XPutImage XGetPixel XPutPixel
Gestionează intrările de la tastatură și mouse	XQueryPointer XTranslateCoordinates XCreateFontCursor XDefineCursor XFreeCursor XUndefineCursor XGetMotionEvents XLookupString XRefreshKeyboardMapping XRebindKeysym
Comunicare între aplicații	XInternAtom XChangeProperty XGetWindowProperty XDeleteProperty XSendEvent XGetSelectionOwner XSetSelectionOwner XConvertSelection

Caracteristici uzuale ale bibliotecii Xlib

Cu toate că biblioteca Xlib dispune de un număr mare de funcții, ele au câteva trăsături comune. Cunoașterea acestora vă poate ajuta să folosiți eficient Xlib.

Trăsăturile comune se încadrează în următoarele categorii:

- Fișiere antet obișnuite
- Convenții de nume pentru funcții și structuri de date
- Ordinea argumentelor în apelurile funcțiilor

Fișiere antet

Dacă folosiți funcții Xlib într-un program, trebuie să includeți următoarele fișiere antet:

```
#include <X11/Xlib.h>
#include <X11/Xutil.h>
```

Primul fișier antet declară funcțiile Xlib și structurile de date. Al doilea fișier antet este folosit de funcțiile utilitare ale bibliotecii Xlib. În plus, veți dori probabil să introduceți alte fișiere antet, precum <X11/cursorfont.h>, dacă folosiți formele standard ale cursorului.

Când un fișier antet este denumit în acest fel (<X11/Xlib.h>), compilatorul se așteaptă să găsească fișierul în subdirectorul X11 al locației standard pentru fișierele antet: directorul /usr/include. Astfel, veți găsi fișierul <X11/Xlib.h> în directorul /usr/include/X11 al sistemului dumneavoastră.

Convenții de nume

Xlib are la bază o convenție a numelor bine definită pentru toate funcțiile, macro-urile și structurile de date. În momentul în care cunoașteți această schemă a denumirilor, puteți intuiti deseori numele funcțiilor și evita erorile uzuale de tastare. Iată care sunt convențiile de nume:

■ *Funcții și macro-uri.* Numele funcțiilor și macrocomenzilor Xlib sunt construite prin concatenarea unuia sau mai multor cuvinte, prima literă a fiecărui cuvânt fiind majusculă (vezi tabelul 23-1 pentru exemple). Numele funcțiilor încep cu un x (mare), dar numele macro-urilor nu încep niciodată cu x. De fapt, dacă o funcție și un macro lucrează identic, numele funcției se formează prin adăugarea prefixului x la numele macro-ului. De exemplu, macro-ul care returnează numele afișării X curente este DisplayString, iar funcția echivalentă este denumită XDisplayString.

■ *Structuri de date.* Structurile de date accesibile utilizatorului sunt denumite exact ca și funcțiile; numele lor încep cu un x (mare). Membrii structurilor de date sunt denumiți cu litere mici, folosind caracterul de subliniere (_) pentru separarea cuvintelor multiple. Structura de date XImage, de exemplu, are un câmp întreg numit bits_per_pixel. Această regulă nu se aplică structurilor de date ai căror membri nu pot fi accesați de către utilizator. Un exemplu în acest sens este structura de date Display, al cărei nume nu începe cu x deoarece utilizatorul nu poate accesa direct funcțiile interne ale acestei structuri.

Puteți folosi aceste convenții pentru a selecta nume pentru propriile dumneavoastră structuri de date și funcții, astfel încât ele să nu intre în conflict cu cele din biblioteca Xlib. Puteți folosi nume cu litere mici pentru variabilele dumneavoastră și nume majuscule pentru macro-uri. Pentru siguranță, puteți decide să adăugați un prefix unic (probabil inițialele organizației sau proiectului dumneavoastră) la toate funcțiile și variabilele dumneavoastră externe.

Ordinea argumentelor în apelurile funcțiilor Xlib

Pe lângă convențiile de nume, funcțiile și macro-urile Xlib își ordonează argumentele într-un mod bine definit. Argumentele apar în ordinea următoare:

- **Afișare.** Afișarea este primul argument al unei funcții sau al unei macrocomenzi.
- **Ferestre, fonturi și alte resurse server X.** Resursele server X – precum fereastră, font și pixmap (hartă de pixeli) – apar imediat după argumentul display. Când sunt folosite câteva resurse, ferestrele și hărțile de pixeli le preced pe toate celelalte. Contextul grafic (GC) apare ultimul dintre resurse.
- **Sursă și destinație.** Multe funcții îndeplinesc sarcini care implică preluarea unei părți din unul sau mai multe argumente (sursa) și stocarea rezultatului în alt argument (destinația). În aceste cazuri, argumentele sursă preced întotdeauna argumentele destinație.
- **x , y , lățime și înălțime.** Multe funcții preiau poziția (x , y) și dimensiunea (lățimea, înălțimea) ferestrei sau a hărții de pixeli ca argumente. Dintre x și y , argumentul x îl precede întotdeauna pe y , în timp ce lățimea este întotdeauna înaintea înălțimii. Când toate cele patru argumente sunt prezente, ordinea este x , y , lățime și înălțime.
- **Bit mask (Mască de biți).** Câteva funcții Xlib schimbă selectiv unul sau mai mulți membri într-o structură. Indicați membrii ce vor fi schimbați prin setarea biților într-o mască de biți (o variabilă de tip întreg în care poziția fiecărui bit corespunde unui membru al structurii). Când o funcție preia o mască de biți ca argument, masca precede întotdeauna pointerul către structură.



Observație

Cunoașterea convenției pentru ordinea argumentelor este chiar mai utilă decât cunoașterea numelor funcțiilor. După ce vă familiarizați cu câteva funcții Xlib, puteți întui deseori lista argumentelor unei funcții, deoarece știți aceste reguli.

Resurse server X

Capitolul 5 folosește termenul *resursă* pentru a se referi la parametrii ce pot fi modificați de utilizator într-o aplicație. Totuși, în contextul server X, *resursa* înseamnă orice este creat la cerere de o aplicație. Astfel, resursele X includ următoarele:

- Fereastră
- Context grafic (GC)
- Font
- Cursor
- Hartă de culori
- Hartă de pixeli

O mare parte a programării X constă în crearea și folosirea acestor resurse, pentru că astfel aplicațiile X sau Motif generează rezultate în ferestre și acceptă intrări de la mouse și tastatură.

Aplicațiile creează resurse apelând funcțiile Xlib. Când programul dumneavoastră creează oricare dintre aceste resurse, funcția Xlib returnează un identificator (ID) al resursei – un identificator pe 32 de biți (de tipul unsigned long în limbajul de programare C). Pentru comoditatea programatorului, fișierul antet `<X11/X.h>` folosește declarația C typedef pentru a defini câteva sinonime – precum Window, Font, Pixmap, Cursor și Colormap – pentru identificatorii resursei. Astfel, atunci când creați o fereastră, vă puteți referi la identificatorul returnat al resursei ca Window.

Ferestre

Dintre resursele X server, ferestrele sunt cele mai importante. Ferestrele reprezintă fundamentul sistemului X Window System. În X, o *fereastră* este o suprafață de pe ecran în

care o aplicație își afișează rezultatele și acceptă intrări (de la tastatură și mouse). X permite imbricarea ferestrelor într-o ierarhie părinte-copil, cu toate ferestrele copil dispuse la marginea ferestrei părinte. Oricând o aplicație X desenează text sau grafică, ea trebuie să specifice o fereastră. De asemenea, toate intrările de la mouse și tastatură sunt asociate cu o fereastră. Xlib conține multe funcții pentru controlul mărimii, culorii și ierarhiei ferestrelor.

Contexte grafice

Pentru a evita trimiterea repetată a atributelor grafice către server, X folosește conceptul de *context grafic* (graphic context – GC) – o resursă server X care deține toate atributele grafice, cum ar fi culorile și fonturile, necesare pentru desenarea într-o fereastră. Aceste atribute grafice controlează aspectul aplicației rezultate. Avantajul este că o aplicație poate crea unul sau mai multe contexte grafice pe server, să le inițializeze și să le folosească mai târziu pentru desenarea într-o fereastră. Deoarece un context grafic este identificat printr-un identificator de resursă, puteți solicita serverului să folosească un set specific de atribute prin includerea unui singur identificator al contextului grafic în cererea de desenare, în locul unui număr variabil de atribute grafice.

Pentru a avea rezultat sub formă de text folosind rutina Xlib de generare a textelor, XDrawString, trebuie să creați contextul grafic și să setați fonturile și culorile înaintea apelării rutinei XDrawString. Puteți folosi rutine Xlib cum ar fi XCreateGC și XChangeGC pentru a crea și manipula contexte grafice.

Fonturi

În X, un *font* desemnează o colecție de bitmap-uri (un șablon numai cu cifrele unu și zero) reprezentând mărimea și forma caracterelor dintr-un set. Serverul X folosește un font pentru a afișa text într-o fereastră. Când apelați o funcție Xlib precum XDrawString pentru a desena unul sau mai multe caractere, serverul încarcă și desenează imaginea corespunzătoare fiecărui caracter din fontul curent.



Referință încrucișată

X oferă un număr mare de fonturi, cu o convenție standard de nume. Trebuie să încărcați unul din aceste fonturi înainte de a desena text într-o fereastră; altfel, serverul folosește un font prestabilit. Pentru a specifica un font, folosiți un identificator de resursă de tipul Font. Capitolul 5 prezintă convenția de nume a fonturilor în sistemul X.

Cursoare

Cursorul reprezintă forma indicatorului de pe ecran care arată poziția curentă a mouse-ului. Pe măsură ce deplasați mouse-ul, cursorul se deplasează pe ecran. Un cursor este similar cu un caracter dintr-un font. De fapt, puteți apela funcția Xlib XCreateFontCursor pentru a crea un nou cursor, selectând unul din caracterele dintr-un font special pentru cursor. La fel ca pentru orice resursă, când creați un cursor obțineți un identificator de resursă de tipul Cursor. După crearea unui cursor, puteți să-l atribuiți unei ferestre folosind funcția XDefineCursor. Dacă nu definiți nici un cursor pentru o fereastră, serverul X folosește cursorul părintelui acesteia.

Schimbarea cursorului este o metodă folositoare pentru a informa utilizatorii despre destinația specială a unei anumite ferestre. Dacă un program de gestionare a ferestrelor vă permite să scalați o fereastră trăgând de colțurile acesteia, de exemplu, atunci el schimbă cursorul atunci când utilizatorul are indicatorul mouse-ului îndreptat spre colțuri.

Hărți de culori

Un monitor X folosește un bloc de memorie, cunoscut ca *buffer cadru* (sau *memorie video*), care este capabil de a stoca un număr fix de biți (de obicei 8) pentru fiecare pixel de pe ecran. Acest număr este așa-numita *valoare a pixelului*. Culoarea afișată la fiecare pixel, pe de altă parte, este rezultatul variației intensității a trei puncte foarte apropiate ale culorilor de bază: roșu (red-R), verde (green-G) și albastru (blue-B). Intensitatea acestor trei componente mai este cunoscută și sub numele de *valoare RGB* sau *triplet RGB*.

Harta de culori (colormap) este cheia generării unui triplet RGB din valoarea unui pixel – ea atribuie valoarea unui pixel unei culori RGB. De exemplu, dacă memoria video stochează valori ale pixelilor pe 8 biți, un pixel poate lua una din cele 2^8 sau 256 de valori posibile. Astfel, harta culorilor trebuie să aibă 256 de intrări, fiecare conținând intensitățile componentelor R, G și B.

Când o aplicație X folosește culori, ea lucrează cu valori ale pixelilor. Pentru a fi siguri că valoarea unui pixel afișează culoarea corectă, aplicația trebuie să identifice harta de culori folosită pentru a transpune valoarea pixelului în culoare. Aplicația face acest lucru folosind harta culorilor.

Plăcile video din majoritatea sistemelor pot folosi numai o hartă a culorilor la un moment dat – o situație ce a condus la conceptul instalării unei hărți de culori. Dacă aplicații multiple instalează hărți de culori în mod independent, rezultatul va fi un haos. Convenția este că aplicațiile X nu trebuie să-și instaleze niciodată propriile hărți de culori; în schimb, ele trebuie să informeze programul de gestionare a ferestrelor despre hărțile de culori de care au nevoie. Dacă primește precizările corecte, programul va instala harta de culori adecvată pentru fiecare aplicație.

Hărți de pixeli

O *hartă de pixeli* (pixmap) este un bloc de memorie din serverul X în care puteți desena exact ca într-o fereastră. De fapt, resursele fereastră și hartă de pixeli sunt cunoscute în X ca desenabile (drawable). Toate funcțiile de desenare acceptă aceste resurse (fereastră și harta de pixeli) ca argumente. Ceea ce desenați într-o hartă de pixeli nu apare pe ecran; pentru a face vizibil conținutul unei hărți de pixeli, trebuie să-l copiați într-o fereastră. Practic, o hartă de pixeli este ca o fereastră ce nu apare pe ecran – un vector bidimensional de valori ale pixelilor ce pot fi utilizate pentru a stoca imagini grafice și modele de hașuri. Dacă fiecare valoare a pixelului dintr-o hartă de pixeli este reprezentată de un singur bit, atunci harta de pixeli este cunoscută ca hartă de biți (bitmap).

Rezumatul evenimentelor X

Orice se întâmplă într-o aplicație X este un răspuns la evenimentele primite de la serverul X. Când o aplicație creează o fereastră și o face vizibilă afișând-o, de exemplu, aplicația nu poate spune dacă serverul a terminat de pregătit fereastra pentru afișare. Pentru a desena în acea fereastră, aplicația trebuie să aștepte un anumit eveniment – un eveniment *Expose* – de la server. Toate datele introduse de utilizator de la mouse și tastatură ajung, de asemenea, la aplicația X sub formă de evenimente.

Deoarece structura elementară a sistemului X nu impune vreun stil particular pentru interfața cu utilizatorul, evenimentele X conțin foarte multe detalii. De exemplu, puteți genera un eveniment X când un buton al mouse-ului este apăsat și altul când acesta este eliberat. Pentru fiecare eveniment, puteți găsi (printre alte lucruri) care buton este implicat, fereastra ce conține cursorul, momentul când se întâmplă evenimentul și coordonatele *x* și *y* ale locației cursorului.

Acest nivel de raportare detaliată a evenimentului permite programatorilor să folosească X pentru a implementa orice tip de interfață cu utilizatorul.

X furnizează 33 de evenimente pentru a trata orice, de la intrări de la mouse și tastatură până la mesaje de la alți clienți X. Tabelul 23-2 rezumă evenimentele X.

Tabelul 23-2 Rezumatul evenimentelor X

Nume eveniment	Semnificație
Evenimente de mouse	
ButtonPress	Butonul mouse-ului apăsat cu indicatorul în fereastră
ButtonRelease	Butonul mouse-ului eliberat cu indicatorul în fereastră
EnterNotify	Indicatorul mouse-ului intră în fereastră
LeaveNotify	Indicatorul mouse-ului iese din fereastră
MotionNotify	Mouse deplasat după oprire
Evenimente de tastatură	
FocusIn	Fereastra este activă pentru introducerea datelor (toate evenimentele de tastatură ulterioare vin în acea fereastră)
FocusOut	Fereastra nu mai este activă pentru introducerea datelor
KeyMapNotify	Are loc după un eveniment EnterNotify sau FocusIn (astfel, serverul X informează aplicația despre starea tastelor după aceste evenimente)
KeyPress	Tastă apăsată (când fereastra este activă)
KeyRelease	Tastă eliberată (când fereastra este activă)
MappingNotify	Tastatura reconfigurată (atribuirea unui șir pentru o tastă s-a schimbat)
Evenimente Expose	
Expose	Fereastra sau o parte a ferestrei acoperită anterior devin vizibile
GraphicsExpose	În timpul operațiilor de copiere a componentelor grafice, părți ale imaginii sursă nu sunt vizibile (ceea ce înseamnă că imaginea copiată nu este completă)
NoExpose	Copierea componentelor grafice s-a încheiat cu succes
Eveniment de notificare a hărții de culori	
ColormapNotify	Harta de culori a ferestrei s-a schimbat
Evenimente de comunicare între clienți	
ClientMessage	Alt client a trimis un mesaj folosind funcția <i>XSends Event</i>
PropertyNotify	Proprietatea asociată cu fereastra s-a schimbat
SelectionClear	Fereastra pierde proprietatea selecției
SelectionNotify	Selecție convertită cu succes
SelectionRequest	Selecția necesită conversie

(continuare)

Tabelul 23-2 Continuare

Nume eveniment	Semnificație
Evenimente de notificare a stării ferestrei	
CirculateNotify	Fereastră plasată mai în față sau mai în spate în cadrul stivei
ConfigureNotify	Fereastră deplasată sau scalată, sau cu poziția schimbată în cadrul stivei
CreateNotify	Fereastră creată
DestroyNotify	Fereastră distrusă
GravityNotify	Fereastră deplasată, deoarece mărimea părintelui ei s-a schimbat
MapNotify	Fereastră afișată
ReparentNotify	Părintele ferestrei s-a modificat
UnmapNotify	Fereastră eliminată de pe ecran
VisibilityNotify	Vizibilitatea ferestrei s-a schimbat (a devenit vizibilă sau invizibilă)
Evenimente de control a structurii ferestrei	
CirculateRequest	Cerere de amplasare mai în față sau mai în spate a ferestrei în cadrul stivei (folosită de programele de gestionare a ferestrelor).
ConfigureRequest	Cerere de deplasare, scalare sau re poziționarea ferestrei în cadrul stivei (folosită de programele de gestionare a ferestrelor)
MapRequest	Fereastră pe care să fie afișată (folosită de programele de gestionare a ferestrelor)
ResizeRequest	Cerere de scalare a ferestrei (folosită de programele de gestionare a ferestrelor)

Cele 33 de evenimente X prezentate în tabelul 23-2 pot fi grupate, în linii mari, în următoarele șapte categorii:

- **Evenimente de mouse.** Serverul X generează evenimente de mouse atunci când utilizatorul apasă un buton al mouse-ului sau deplasează mouse-ul.
- **Evenimente de tastatură.** Serverul generează evenimente de tastatură când utilizatorul apasă sau eliberează orice tastă. Aceste evenimente sunt furnizate unei aplicații numai dacă o fereastră a aplicației este activă pentru introducerea de date. De obicei, programul de gestionare a ferestrelor decide cum este transferat focus-ul de la o fereastră la alta. Există două modele uzuale de focalizare: executarea unui clic într-o fereastră pentru a tasta în ea (folosit de Macintosh și Microsoft Windows), sau permițând focalizării să urmărească indicatorul mouse-ului (ceea ce înseamnă că fereastră ce conține indicatorul mouse-ului este activă).
- **Evenimente Expose.** Dintre toate evenimentele X, un eveniment Expose este cel mai important; aplicațiile desenează în ferestrele lor ca răspuns la acest eveniment. Aproape toate aplicațiile X solicită și procesează acest eveniment. (În aplicațiile Motif, cele mai multe evenimente X sunt manevrate în culise, dar dumneavoastră trebuie să aveți grijă de evenimentele Expose în primitivele DrawingArea.) Evenimentele GraphicsExpose și NoExpose presupun copierea dintr-o parte în alta a unei ferestre sau hărți de pixeli. Aceste evenimente permit aplicațiilor să rezolve cazurile în care sursa operației de copiere este acoperită de altă fereastră, iar conținutul suprafeței care nu este vizibilă nu este disponibil pentru copiere.

- **Eveniment de notificare a hărții de culori.** Serverul generează un eveniment ColorMapNotify de fiecare dată când o aplicație schimbă harta de culori asociată cu o fereastră sau instalează o nouă hartă de culori. Aplicațiile X care funcționează corect ar trebui să gestioneze schimbările hărții de culori prin intermediul programului de gestionare a ferestrelor.
- **Evenimente de comunicare între clienți.** Aceste evenimente trimit informații de la o aplicație X la alta. Conceptele de proprietate și selecție sunt folosite în acest scop.
- **Evenimente de notificare a stării ferestrei.** Serverul generează aceste evenimente de fiecare dată când o fereastră este deplasată sau scalată sau când poziția acesteia în cadrul stivei este modificată. Aceste evenimente sunt folosite pentru a ține evidența schimbărilor în ceea ce privește amplasarea ferestrelor pe ecran. De obicei, programele de gestionare a ferestrelor folosesc aceste evenimente în scopul precizat; aplicația dumneavoastră poate să le folosească, de asemenea, dacă doriți să modificați mărimea și poziția ferestrelor inferioare când utilizatorul scalează din nou fereastra cea mai de sus.
- **Evenimente pentru controlul structurii ferestrei.** Aceste evenimente sunt folosite aproape exclusiv de către programele de gestionare a ferestrelor pentru a intercepta încercarea unei aplicații de a-și schimba aspectul ferestrelor. Prin monitorizarea evenimentului MapRequest, de exemplu, programul de gestionare a ferestrelor poate sesiza când o aplicație își mapează fereastra cea mai de sus. Când se întâmplă acest lucru, programul de gestionare a ferestrelor poate adăuga propria lui margine ferestrei și o poate plasa într-o poziție adecvată pe ecran.

Elemente de programare Xlib

Setul de instrumente Motif are grijă de multe detalii referitoare la crearea ferestrelor și tratarea evenimentelor atunci când scrieți aplicații Motif. Utilizatorul furnizează de obicei funcții de apel invers, care realizează sarcini specifice aplicațiilor. Aveți nevoie doar de un set mic de funcții Xlib într-o aplicație Motif. Următoarele secțiuni vă oferă o imagine generală despre tipurile de programare Xlib pe care trebuie să le realizați într-o aplicație Motif.

Setarea formei și culorii cursorului

Cursorul determină apariția pe ecran a indicatorului mouse-ului. În X, un cursor este definit de următorii parametri:

- Harta de biți sursă
- Masca hărții de biți
- Culorile pentru prim-plan și fundal, specificate ca valori RGB
- Punctul fierbinte (hotspot) (punctul din harta de biți a cursorului ce definește locația indicatorului pe ecran)

Hărțile de biți sunt masive rectangulare mici formate numai din cifrele unu și zero (de obicei, 16x16 sau 32x32). Când este desenat cursorul, serverul X colorează pixelii corespunzători cifrei unu folosind culoarea pentru prim-plan, în timp ce pixelii corespunzători cifrei zero apar în culoarea fundalului. Masca hărții de biți determină conturul în care este desenată forma cursorului. Punctul fierbinte determină locația indicatorului. Pentru multe forme de cursor, punctul fierbinte (hotspot) se află în centrul hărții de biți a cursorului. Pentru un cursor în formă de săgeată, punctul fierbinte este vârful săgeții.

Puteți atribui un cursor oricărei ferestre din aplicația dumneavoastră Motif. Următorul exemplu vă prezintă crearea unui cursor nou dintr-un font de cursor standard și atribuirea acestuia ferestrei unei primitive DrawingArea dintr-o aplicație Motif:

```
# include <X11/cursorfont.h>
Cursor xhair_cursor;
.
.
/* Creeaza un cursor sub forma a doua fire rectangulare pentru suprafata de
desenare. Considerati ca
"drawing_area"
* este numele primitivei DrawingArea.
*/
xhair_cursor = XCreateFontCursor(XtDisplay(drawing_area),
XC_crosshair);
.
.
/* Modifica forma cursorului din suprafata de desenare (apelati-o dupa ce
* primitivele sunt realizate.
*/
/* Realizeaza toate primitivele */
XtRealizeWidget(top_level);

/* Seteaza cursorul pentru suprafata de desenare a ferestrei primitivei */
XDefineCursor(XtDisplay(drawing_area), XtWindow(drawing_area),
xhair_cursor);

/* Elibereaza cursorul */
XFreeCursor(XtDisplay(drawing_area), xhair_cursor);
```

După ce ați făcut acest lucru, forma cursorului se schimbă într-un dreptunghi cu margini subțiri ori de câte ori indicatorul intră în fereastra `my_window`. Această selecție rămâne valabilă până când renunțați la definirea cursorului pentru acea fereastră, apelând funcția `XUndefineCursor`. Când renunțați la definirea unui cursor asociat unei ferestre, serverul afișează cursorul părintelui acelei ferestre atunci când indicatorul se află în fereastra ce are cursorul nedefinit.

Observați că al doilea argument al funcției `XCreateFontCursor` specifică forma cursorului cu un nume simbolic. Aceste nume sunt definite în fișierul antet `<X11/cursorfont.h>`.

După ce atribuiți un cursor unei ferestre, dacă nu intenționați să vă mai referiți la aceasta, puteți elibera cursorul apelând funcția `XFreeCursor`. Orice fereastră care are atribuit acest cursor va continua să-l afișeze; serverul va elimina cursorul numai după ce acesta nu va mai fi definit în nici o fereastră. După ce ștergeți definiția unui cursor, nu trebuie să vă mai referiți din nou la identificatorul său.

Când este creat un cursor, acesta are, în mod prestabilit, un prim-plan negru și un fundal alb. Pentru a schimba culoarea unui cursor, folosiți funcția `XRecolorCursor` astfel:

```
XColor fgcolor, bgcolor; /* Culori în structura XColor */
Cursor arrow_cursor; /* Cursor a carui culoare este stabilită */
```

```
XRecolorCursor(theDisplay, arrow_cursor, &fgcolor, &bgcolor);
```

Trebuie să alocați culorile pentru prim-plan și fundal înainte de a le folosi în apelul funcției `XRecolorCursor`.

Puteți, de asemenea, să vă folosiți propria hartă de biți sursă și masca pentru harta de biți pentru a defini un cursor personalizat. După ce aveți cele două hărți de pixeli (hărțile de biți sunt hărți de pixeli de adâncime 1), puteți folosi funcția `XCreatePixmapCursor` pentru a crea un nou cursor. Această funcție are nevoie de cele două hărți de pixeli, de culorile pentru prim-plan și fundal și de coordonatele punctului fierbinte, așa cum se vede în următorul exemplu:

```
Display *theDisplay;
Cursor my_cursor;
```

```
Pixmap source, mask;
XColor fgcolor, bgcolor;
unsigned int x_hot, y_hot;
```

```
my_cursor = XCreatePixmapCursor(theDisplay, source, mask
&fgcolor, &bgcolor, x_hot, y_hot);
```

Altă modalitate de a obține un cursor este de a selecta un anumit caracter dintr-un font și utilizarea hărții de biți a acelui caracter drept cursor. Înainte de a folosi fontul, trebuie să-l încărcați apelând funcția `XLoadFont`. Apoi puteți crea cursorul folosind funcția `XCreateGlyphCursor`.

Desenarea graficii și a textului

Pentru a desena grafică sau text într-o fereastră X, trebuie să creați un *context grafic* – o structură de date (resursă) în care serverul X stochează atribute grafice cum ar fi culorile pentru fundal și prim-plan, tipul de linie și fontul. Aspectul graficii și al textului sunt controlate de aceste atribute.

Crearea unui context grafic

Pentru a crea un context grafic, folosiți funcția `XCreateGC` din biblioteca Xlib. Această funcție preia patru argumente, în ordinea următoare:

- Un pointer către afișarea X
- Identificatorul unei desenabile (o variabilă `Window` sau `Pixmap`)
- O mască de biți de tipul *unsigned long* ce indică atributele contextului grafic pe care doriți să le specificați.
- Adresa unei structuri `XGCValues`

Specificați diferite atribute grafice în structura `XGCValues` și folosiți masca de biți pentru a indica membrii structurii cu nume valide. Iată un exemplu de apelare a funcției `XCreateGC`:

```
Display *disp;
Drawable win;
unsigned long mask;
XGCValues xgcv;
GC gc;
```

```
/* Seteaza valorile membrilor selectati ai structurii xgcv.
* Apoi seteaza masca. Acest exemplu seteaza pixelii pentru prim-plan si
* fundal la culorile prestabilite alb si negru pentru ecran.
*/
```

```
xgcv.foreground = WhitePixel(disp, DefaultScreen(disp));
xgcv.background = BlackPixel(disp, DefaultScreen(disp));
mask = GCForeground | GCBackground;
```

```
gc = XCreateGC(disp, win, mask, &xgcv);
```

Deoarece contextul grafic este o resursă, el consumă memorie din serverul X. Prin urmare, trebuie să eliberați un context grafic atunci când nu mai aveți nevoie de el. Puteți face acest lucru cu funcția `XFreeGC`, astfel:

```
Display *disp;
GC gc;
```

```
XFreeGC(disp, gc);
```

Serverul X eliberează automat toate resursele aplicației dumneavoastră (inclusiv contextele grafice) când aplicația se încheie, astfel că trebuie să distrugeți în mod explicit un context grafic numai atunci când ați creat unul pentru un scop temporar.

Atribute ale contextului grafic

Serverul X se ocupă de întreținerea contextelor grafice, astfel că nu este nevoie să cunoașteți detalii interne despre acestea. Specificați valorile unui context grafic printr-o structură `XGCValues`. Ideea de bază este de a seta valorile atributelor pe care le doriți și apoi să creați o mască de biți pentru a indica atributele specificate. Serverul X folosește structura `XGCValues` împreună cu masca de biți pentru a determina componentele contextului grafic ce urmează a fi modificate.

Definiția structurii `XGCValues` vă oferă o idee despre atributele grafice pe care le puteți controla. Iată cum este definită această structură în fișierul antet `<X11/Xlib.h>`:

```
typedef struct
{
    int      function;          /* Operatie cu pixeli */
    unsigned long plane_mask;  /* Planuri de biti afectate */
    unsigned long foreground;  /* Valoarea pixelului pentru prim-plan */
    unsigned long background; /* Valoarea pixelului pentru fundal */
    int      line_width;       /* Grosimea liniei (0 sau mai mult) */
    int      line_style;       /* Unul dintre: LineSolid, LineOnOffDash,
    LineDoubleDash */
    int      cap_style;        /* Unul dintre: CapNotLast, CapButt, CapRound,
    CapProjecting */
    int      join_style;       /* Unul dintre: JoinMiter, JoinRound, JoinBevel */
    int      fill_style;       /* Unul dintre: FillSolid, FillTiled, FillStippled
    FillOpaqueStippled */
    int      fill_rule;        /* Unul dintre: EvenOddRule, WindingRule */
    int      arc_mode;         /* Unul dintre: ArcChord, ArcPieSlice */
    Pixmap   tile;            /* Harta de pixeli pentru acoperire */
    Pixmap   stipple;         /* Harta de biti pentru gravare cu puncte */
    int      ts_x_origin;      /* Distanțele pe x si y pentru acoperire */
    int      ts_y_origin;      /* sau operatii de gravare cu puncte */
    Font      font;           /* Fontul prestabilit */
    int      subwindow_mode;   /* Unul dintre: ClipByChildren, IncludeInferiors */
    Bool      graphics_exposures; /* True=genereaza prezentari grafice */
    int      clip_x_origin;    /* Originea mastii decupate */
    int      clip_y_origin;    /* Originea mastii decupate */
    Pixmap   clip_mask;       /* Harta de biti pentru decupare */
    int      dash_offset;     /* Controleaza liniile intrerupte */
    char      dashes;         /* Model de hasuri */
} XGCValues;
```

Pentru a seta aceste atribute, aveți nevoie, de asemenea, de numele măștii de biți asociate cu fiecare membru al structurii `XGCValues`. Tabelul 23-3 listează măștile de biți folosite pentru a selecta anumiți membri ai structurii `XGCValues`. Când setați atribute multiple, folosiți o combinație de măști care este rezultatul operației SAU pe biți (indicat prin operatorul `C`) corespunzătoare atributelor pe care vreți să le setați.

Tabelul 23-3 conține valoarea prestabilită a fiecărui atribut. Când creați un context grafic fără să specificați vreo valoare pentru atribute, serverul X inițializează contextul grafic cu valorile prestabilite ale atributelor.

Tabelul 23-3 Constante ale măștii de biți pentru atribute GC

Nume atribut	Constantă mască de biți	Descriere
function	GCFFunction	Operații cu pixeli (Prestabilit: suprascrisce pixelii existenți)
plane_mask	GCPlaneMask	Planuri de biți afectate (Prestabilit: toate planurile de biți afectate)

Nume atribut	Constantă mască de biți	Descriere
foreground	GCForeground	Pixel de prim-plan (Prestabilit: 0)
background	GCBBackground	Pixel de fundal (Prestabilit: 1)
line_width	GCLineWidth	Grosimea liniei (Prestabilit: 0)
line_style	GCLineStyle	Stilul liniei (Prestabilit: linie solidă)
cap_style	GCCapStyle	Cum se termină liniile (Prestabilit: se termină în punctul de sfârșit fără nici o proiecție)
join_style	GCJoinStyle	Cum se unesc liniile (Prestabilit: îmbinare în unghi ascuțit)
fill_style	GCFillStyle	Stilul de umplere (Prestabilit: umplere uniformă)
fill_rule	GCFillRule	Cum sunt umplute cifrele (Prestabilit: umplere folosind regula par-impar, în care un punct este în interior dacă o linie trasă din exteriorul cifrei intersectează marginile acesteia de un număr impar de ori)
arc_mode	GCArcMode	Aspectul arcelor umplute ca felii de tort sau închise cu o coardă (Prestabilit: arce umplute ca felii de tort)
tile	GCTile	Hartă de pixeli pentru acoperire (Prestabilit: hartă de biți umplută cu pixeli de prim-plan).
stipple	GCStipple	Hartă de biți pentru gravare cu puncte (Prestabilit: hartă de biți numai cu cifre 1). Modelul de gravare cu puncte este ca un șablon prin care culoarea de prim-plan este aplicată unei desenabile.
ts_x_origin	GCTileStipXOrigin	Deplasament pe axa x pentru acoperire sau gravare cu puncte (Prestabilit: 0)
ts_y_origin	GCTileStipYOrigin	Deplasament pe axa y pentru acoperire sau gravare cu puncte (Prestabilit: 0)
font	GCFont	Fontul prestabilit pentru afișarea textului (Prestabilit: depinde de serverul X)
subwindow_mode	GCSubwindowMode	Desenează sau nu în ferestre copil (Prestabilit: nu desenează în ferestre copil)
graphics_exposures	GCGraphicsExposures	Evenimente grafice de prezentare generate dacă constanta este True (Prestabilit: True)
clip_x_origin	GCClipXOrigin	Originea x pentru clip_mask (Prestabilit: 0)
clip_y_origin	GCClipYOrigin	Originea y pentru clip_mask (Prestabilit: 0)
clip_mask	GCClipMask	Hartă de biți pentru decupare (Prestabilit: nici o mască pentru lipire folosită)
dash_offset	GCDashOffset	Punctul de pornire în modelul de umplere cu linii întrerupte (Prestabilit: 0)
dashes	GCDashList	Model de umplere cu linii întrerupte (Prestabilit: model cu 4 pixeli activați și 4 neactivați).

Desenarea punctelor

Cea mai simplă operație grafică în X este desenarea unui punct într-o fereastră sau într-o hartă de pixeli. Puteți desena un singur punct având coordonatele x și y cu următorul apel:

```
Display disp; /* Conexiunea la serverul X */
Window win; /* Suprafata de desenare - - o fereastră */
GC thisGC; /* Context grafic pentru desen */
int x, y; /* Punct ce va fi desenat */
```

```
XDrawPoint(disp, win, thisGC, x, y);
```

Pixelul de la locația respectivă este setat pe culoarea prim-planului specificată în contextul grafic. Alte atribute din contextul grafic controlează aspectul final al punctului. De exemplu, punctul nu este desenat dacă se află în afara măștii decupate.



Sugestie

Dacă doriți să desenați un număr mare de puncte, toate folosind același context grafic, puteți folosi funcția `XDrawPoints` pentru a le desena pe toate în același timp cu o singură cerere de protocol X. Stocați punctele într-un șir de structuri `Xpoint`.

Structura `XPoint` este definită în fișierul antet `<X11/Xlib.h>` astfel:

```
typedef struct
{
    short x, y; /* coordonatele x si y ale punctului */
} Xpoint;
```

Apelați funcția `XDrawPoints` în maniera obișnuită, cu o afișare, o variabilă de desenare și un context grafic drept primele trei argumente:

```
XPoint pt[10];
int numpt = 10;
...
XDrawPoints(p_disp, window_1, thisGC, pt, numpt,
            CoordModeOrigin);
```

Masivul de puncte și numărul de puncte urmează după cele trei argumente standard. Ultimul argument indică serverului cum să interpreteze coordonatele punctelor din șir. Puteți specifica una din următoarele constante:

- `CoordModeOrigin` înseamnă că coordonatele sunt relative față de originea ferestrei sau a hărții de pixeli.
- `CoordModePrevious` este folosită atunci când coordonatele fiecărui punct sunt exprimate sub formă de distanțe pe x și y față de punctul precedent. Primul punct se presupune că este relativ la originea ferestrei.

Desenarea liniilor

Xlib conține trei funcții de desenare a liniilor:

- `XDrawLine`
- `XDrawSegments`
- `XDrawLines`

Sintaxa de apelare a funcției `XDrawLine` este următoarea:

```
XDrawLine(disp, win, thisGC, x1, y1, x2, y2);
```

Această funcție desenează o linie între punctele $(x1, y1)$ și $(x2, y2)$ în interiorul desenatului numită `win`, folosind atributele de linie specificate în `GC (thisGC)`.

`XDrawSegments` desenează câteva segmente de dreaptă separate, folosind aceleași atribute grafice. Segmentele sunt specificate prin intermediul structurii `XSegment`, care este definită în fișierul antet `<X11/Xlib.h>` astfel:

```
typedef struct
{
    short x1, y1; /* Coordonatele punctului de inceput al segmentului */
    short x2, y2; /* Coordonatele punctului de sfarsit al segmentului */
} Xsegment;
```

Următorul exemplu arată cum se folosește funcția `XDrawSegments` pentru a desena câteva segmente de dreaptă:

```
Display *disp;
Window window_1;
GC thisGC;
XSegment lines[ ] =
{
    {50, 80, 150, 200},
    {10, 20, 35, 60},
    {250, 10, 200, 100}
};
int numsegs = sizeof(lines) / sizeof(XSegment);
```

```
XDrawSegments(disp, window_1, thisGC, lines, numsegs);
```

Funcția `XDrawLines` este asemănătoare cu `XDrawPoints`, cu o singură diferență: `XDrawPoints` desenează puncte, în timp ce `XDrawLines` le unește cu o linie. Apelați funcția `XDrawLines` cu aceleași argumente pe care le folosiți pentru `XDrawPoints`.

Desenarea și umplerea dreptunghiurilor

Biblioteca Xlib conține câteva funcții pentru desenarea dreptunghiurilor. Funcția `XdrawRectangle` este folosită pentru desenarea unui dreptunghi, atunci când se cunosc coordonatele colțului din stânga-sus precum și lățimea și lungimea acestuia. Apelarea funcției se face astfel:

```
XDrawRectangle (disp, window, thisGC, x, y, width, height);
```

Pentru a desena conturul dreptunghiului, această funcție desenează următoarele linii:

- (x, y) to $(x+width, y)$
- $(x+width, y)$ to $(x+width, y+height)$
- $(x+width, y+height)$ to $(x, y+height)$
- $(x, y+height)$ to (x, y)

Puteți folosi același context grafic (GC) pentru a desena mai multe dreptunghiuri apelând funcția `XDrawRectangles`, care preia un masiv de dreptunghiuri și numărul lor ca argumente, astfel:

```
XRectangle rects [ ]; /* Sir de dreptunghiuri */
int nrects; /* Numar de dreptunghiuri */
```

```
XDrawRectangles(disp, window, thisGC, rects, nrects)
```

Funcția `XRectangle` este o structură pentru stocarea parametrilor unui dreptunghi. Ea este definită în fișierul antet `<X11/Xlib.h>` astfel:

```
typedef struct
{
    short x, y; /* Coltul din stanga-sus */
```

```
unsigned short width, height; /* Latimea si inaltimea */
} Xrectangle;
```

Puteți desena un dreptunghi umplut apelând funcția `XFillRectangle`. Apelați această funcție la fel ca și `XDrawRectangle`, folosind exact aceleași argumente. Observați că atunci când construiți un dreptunghi umplut, lățimea și înălțimea suprafeței umplute sunt exact cele specificate în apelul funcției `XFillRectangle`.

`XFillRectangle` este funcția pentru desenarea dreptunghiurilor multiple umplute. Funcția este analoagă cu `XDrawRectangles` și este apelată cu aceleași argumente.

Desenarea poligoanelor

Un *poligon* este o figură mărginită de mai multe linii. Pentru a desena chenarul unui poligon, folosiți funcția `XDrawLines`. Pentru poligoane umplute, biblioteca `Xlib` furnizează funcția `XFillPolygon`, care se folosește astfel:

```
int shape; /* Una dintre: Convex, NonConvex, or Complex */
int mode; /* Una dintre: CoordModeOrigin or
            CoordModePrevious */
XPoint points [ ] /* Vertexurile poligonului */
int numpoints; /* Numarul de vertexuri */
```

```
XFillPolygon(dis, win, thisGC, points, numpoints,
             shape, mode);
```

Trebuie să specificați vertexurile poligonului într-un șir de structuri `XPoint`, exact cum faceți când desenați puncte multiple cu `XDrawPoints`. De asemenea, argumentul `mode` este interpretat la fel ca și în cazul funcției `XDrawPoints`.

Argumentul `shape` ajută serverul să optimizeze algoritmul de umplere. Specificați `Convex` ca argument numai dacă poligonul dumneavoastră are o asemenea formă încât o linie ce unește oricare două puncte din interiorul poligonului nu are nici un punct în afara poligonului. (De exemplu, triunghiurile și dreptunghiurile sunt forme convexe.)

Dacă forma nu este convexă, iar nici una din laturi nu se intersectează cu celelalte, ar trebui să folosiți `NonConvex` ca argument al formei. Pentru poligoane ca laturi care se intersectează, folosiți `Complex`. În cazul în care nu sunteți siguri de forma unui poligon, puteți specifica fără probleme `Complex` drept formă. Procesul de desenare va fi puțin mai lent, dar rezultatul va fi corect.

Regula de umplere din contextul grafic determină care puncte sunt umplute de `XFillPolygon`. Puteți seta regula de umplere apelând funcția `XSetFillRule`.

Desenarea arcelor, cercurilor și elipselor

În X, arcele, elipsele și cercurile sunt manipulate de funcțiile de desenare a arcelor: `XDrawArc` și `XDrawArcs`. Prima funcție este pentru un singur arc, iar a doua pentru mai multe arce. Apelați funcția `XDrawArc` astfel:

```
XDrawArc(dis, window, gc, x, y, width, height, angle1, angle2);
```

Puteți considera un arc ca o porțiune a unei elipse. Desenarea unui arc implică specificarea *dreptunghiului de încadrare*, care este cel mai mic dreptunghi care încadrează complet elipsa de care aparține arcul. Specificați dreptunghiul prin coordonatele (x,y) ale colțului din stânga-sus și prin dimensiunile acestuia (width și height). Indicați unghiul la care începe arcul (angle1) precum și unghiul pe care-l acoperă (angle 2).

Serverul X desenează arcul pornind dintr-un punct de pe elipsă situat de-a lungul liniei ce face unghiul `angle1`, măsurat în sens trigonometric, cu 0 grade în dreptul poziției acului orar al unui

ceas care indică ora 3. Apoi serverul urmărește conturul elipsei în direcție trigonometrică, până ce acoperă unghiul specificat de `angle2`.

Unghiurile `angle1` și `angle2` sunt valori întregi, care specifică unghiurile în multipli de 1/64 dintr-un grad. Astfel, pentru a desena un arc de 60 de grade începând de la unghiul de 30 de grade față de direcția de zero, trebuie să folosiți următorul apel:

```
XDrawArc(p_disp, window, thisGC, x, y, width, height, 30*64, 60*64);
```

Puteți desena o elipsă pornind de la zero grade și specificând un arc de 360*64. Dacă lățimea și lungimea dreptunghiului de încadrare sunt egale, obțineți un cerc.

Desenarea textului

Pentru a afișa text, trebuie să selectați mai întâi un font. În X, fonturile sunt resurse localizate pe server. Aplicațiile trebuie să încarce un font înainte de a-l folosi. Când un server X încarcă un font cu succes, el returnează un identificator de resursă pe care aplicația îl folosește ulterior pentru a se referi la acel font. În capitolul 5 ați învățat cum să denumiți un font. Când numele este cunoscut, puteți încărca un font apelând funcția `XLoadFont` astfel:

```
Display disp; /* Identifica conexiunea cu serverul X */
char fontname [ ] = "helvetica-bold r*140"; /* Numele fontului */
Font helvb14; /* Identificatorul fontului */

if ((helvb14 = XLoadFont(dis, fontname)) == None)
{
    fprintf(stderr, "Cannot load font: %s\n", fontname);
    /* Tratarea erorii. Probabil ca foloseste doar fontul prestabilit */
    /* ... */
}
```

Când funcția `XLoadFont` returnează o valoare diferită de zero (ceea ce înseamnă că funcția a rulat cu succes), puteți să începeți să folosiți fontul Helvetica Bold de 14 puncte, prezentat în exemplu, prin setarea atributului de font pentru un context grafic. Pentru a face acest lucru, apelați funcția `XSetFont` astfel:

```
XSetFont(dis, thisGC, helvb14);
```

Ulterior, când se desenează text cu acest context grafic, acesta va utiliza fontul Helvetica Bold de 14 puncte.

Deoarece fonturile sunt resurse localizate pe server, când aplicația dumneavoastră nu mai are nevoie de un font, ar trebui să-l eliberați cu funcția `XUnloadFont`. Apelul acestei funcții are următoarea formă:

```
XUnloadFont(p_disp, helvb14);
```

Remarcați că, dacă aveți un context grafic cu `helvb14` drept font și descărcați acest font, serverul X nu-l descarcă până când acel context grafic nu este distrus.

După ce dispuneți de un context grafic cu fontul adecvat, aplicația dumneavoastră poate apela una din următoarele funcții pentru a afișa text:

```
■ XDrawString(Display* display, Drawable d, GC gc, int x, int y, const char*
string, int length); – Desenează text folosind numai culoarea pentru prim-plan.
```

```
■ XDrawImageString(Display* display, Drawable d, GC gc, int x, int y, const char*
string, int length); – Desenează caractere folosind atât culori de prim-plan cât și de fundal.
```

```
■ XDrawText(Display* display, Drawable d, GC gc, int x, int y, XTextItem* items,
int nitems); – Desenează câteva șiruri text pe o linie.
```

Fiecare funcție desenează câteva caractere pe o singură linie. Primele două funcții desenează toate caracterele folosind fontul specificat ca argument în contextul grafic. Funcțiile sunt apelate în același mod, astfel:

```
char string [ ]; /* Sir ce va fi afisat */
int nchars; /* Numarul caracterelor din sir */
```

```
XDrawString(dispatch, window, thisGC, x, y, string, nchars);
XDrawImageString(dispatch, window, thisGC, x, y, string, nchars);]
```

Specificați o poziție de pornire, unde serverul plasează originea hârtii de biți a primului caracter. Apoi serverul copiază valoarea pixelului de prim-plan (din contextul grafic) în toți pixelii corespunzători cifrelor unu din harta de biți. `XDrawString` nu modifică pixelii acolo unde harta de biți este zero. `XDrawImageString`, pe de altă parte, umple, de asemenea, pixelii corespunzători cifrelor zero din fiecare hartă de biți a caracterului folosind culoarea de fundal a contextului grafic (folosind gravarea sau acoperirea, dacă există vreuna).



Dacă doriți să folosiți mai mult de un font pe aceeași linie de text, folosiți funcția `XDrawText`. Această funcție acceptă informații despre segmentele de șir dintr-o structură `XTextItem`, care este definită în fișierul antet `<X11/Xlib.h>` după cum

Sugestie urmează:

```
typedef struct
{
    char chars; /* Pointer catre sirul ce va fi desenat */
    int nchars; /* Numarul de caractere din sir */
    int delta; /* Distanța de la ultimul caracter al sirului anterior */
    Font font; /* Font = None înseamnă folosirea fontului contextului grafic */
} XTextItem;
```

Fiecare structură `XTextItem` conține informații despre un singur bloc de text. Primii doi membri identifică șirul și lungimea lui. Membrul `delta` este o distanță măsurată, în pixeli, aplicată înainte de desenarea acestui șir. Ultimul membru specifică fontul ce va fi utilizat pentru desenarea acestui șir. Când apelați funcția `XDrawText`, trebuie să furnizați, ca de obicei, pointerul `Display`, suprafața de desenare și contextul grafic, urmat de locația unde trebuie să apară șirul și de șirurile dintr-un vector de structuri `XTextItem`, după cum urmează:

```
XTextItem text_chunks [ ]; /* Vector de siruri ce vor fi afisate */
int numchunks; /* Numarul structurilor XTextItem */
```

```
/* Initializati mai intai „text_chunks” ... */
```

```
XDrawText(dispatch, window, thisGC, x, y, text_chunks, numchunks);
```

`XDrawText` afișează fiecare șir folosind fontul specificat în câmpul `font` din structura `XTextItem` corespunzătoare. Funcția execută această operație încărcând fontul în contextul grafic pe care-l indicați în apelul funcției. Dacă în structura `XTextItem` câmpul `font` este setat pe `None`, funcția `XDrawText` folosește fontul existent în contextul grafic.

Folosirea funcțiilor de desenare în Motif

Pentru a folosi funcțiile bibliotecii `Xlib` în orice aplicație, aveți nevoie de pointerul `Display`, de identificatorul ferestrei, iar pentru funcții de desenare de un context grafic (`GC`). `Xt Intrinsics` (fundatia pe care este construit Motif) vă oferă macro-uri și funcții pentru a obține acești parametri pentru orice primitivă.

Pointerul Display și identificatorul ferestrei

Aplicațiile Motif lucrează cu primitive, dar cele mai multe funcții `Xlib` necesită un pointer către structura `Display` și un identificator de fereastră ca argumente. Cunoșcând identificatorul unei primitive, puteți obține pointerul către structura proprie `Display` folosind funcția `XtDisplay`. Asemănător, funcția `XtWindow` returnează identificatorul ferestrei asociată unei primitive.

Să presupunem că doriți să utilizați funcția `XCclearWindow` din biblioteca `Xlib` pentru a șterge fereastra unei primitive. Cunoșcând identificatorul primitivei `w`, puteți realiza această operație cu următorul fragment de cod:

```
# include <X11/Intrinsics.h>
```

```
Widget w;
Display *p_disp;
Window win;
```

```
p_disp = XtDisplay(w);
win = XtWindow(w);
XCclearWindow(p_disp, win);
```

Identificatorul ferestrei returnat de `XtWindow` va fi `NULL` dacă primitiva nu a fost realizată.

Pointerul `Display` totuși, este valid imediat după ce este creată primitiva dacă apelați funcția `XtCreateWidget` sau altă funcție echivalentă.

Crearea contextelor grafice în Motif

Pentru a desena text și grafică în fereastra unei primitive, trebuie să creați unul sau mai multe contexte grafice. În cazul în care utilizați numai biblioteca `Xlib`, folosiți funcții precum `XCreateGC`, `XCopyGC` și `XChangeGC` pentru a crea și manipula contexte grafice. `Xt Intrinsics` furnizează funcția `XtGetGC` pentru crearea contextelor grafice. Această funcție încearcă să minimizeze numărul contextelor grafice create, ținând evidența acestora în toate primitivele dintr-o aplicație. `Xt Intrinsics` creează un nou context grafic numai atunci când nici unul din cele existente nu are atribute adecvate cererii dumneavoastră din apelul `XtGetGC`.

Când creați un context grafic pentru o primitivă, ar trebui să preluați valorile pixelilor de prim-plan și fundal din resursele primitivei. Astfel, veți folosi culorile de prim-plan și fundal pe care utilizatorul le-a specificat pentru acea primitivă într-un fișier resursă.

Valorile resurselor unei primitive pot fi obținute apelând funcția `XtGetValues`. Pașii sunt asemănători celor folosiți pentru setarea valorilor resurselor. Să presupunem că doriți culorile pentru prim-plan și fundal ale primitivei `DrawingArea` numită `drawing_area`. Următoarele exemple vă arată cum puteți obține aceste valori și cum puteți crea un context grafic cu aceste atribute:

```
Arg args [20];
Cardinal nargs;
Widget drawing_area;
XGCValues xgcv;
GC theGC;
int fg, bg;
```

```
/* Preia culorile de prim-plan si fundal din resursele primitivei
 * colors from the widget's resources.
 */
```

```
nargs = 0;
XtSetArg( args [nargs], XmNforeground, &fg); nargs++;
XtSetArg( args [nargs], XmNbackground, &bg); nargs++;
XtGetValues(drawing_area, args, nargs);
```

```
/* Acum, definiți un context grafic cu aceste culori. */
xgcv.foreground = fg;
xgcv.background = bg;
theGC = XtGetGC(drawing_area, GCForeground | GCBackground,
&Xgcv);
```

Observați că, atunci când obțineți valoarea unei resurse, furnizați adresa unei variabile în care funcția XtGetValues plasează valorile.

După ce creați un context grafic (GC), îl puteți manipula folosind funcțiile Xlib. Remarcați totuși că contextul grafic returnat de funcția XtGetGC poate fi numai citit, deci nu-l puteți modifica. Folosiți funcția XCreateGC dacă aveți nevoie de un context grafic pe care să-l puteți modifica.

Un program Motif de desenare a liniilor

Următorul listing prezintă programul Motif `xm1ines` de desenare a liniilor. Programul poate desena figuri formate din linii frânte. Când utilizatorul execută clic pentru prima dată pe butonul stâng al mouse-ului în suprafața de desenare, este marcat un colț al liniei. Pe măsură ce utilizatorul deplasează mouse-ul ținând butonul apăsat, pe ecran apare linia în direcția deplasării mouse-ului. Linia finală este desenată atunci când utilizatorul eliberează butonul.

```
/*-----*/
/* Fisier: xm1ines.c
*
* Un program Motif care desenează linii.
*/
/*-----*/
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/cursorfont.h>

#include <Xm/Xm.h>
#include <Xm/RowColumn.h>
#include <Xm/MainW.h>
#include <Xm/DrawingA.h>

#define MAXARGS 20
#define MAXLINES 100
#define WIDTH 400
#define HEIGHT 300
static char message [ ] =
    „Țineți apăsat butonul stâng al mouse-ului, deplasați-l și apoi eliberați butonul.”;
static int msglen = XtNumber(message) - 1;

/* Sir de linii */

Xsegment lines[MAXLINES];
int numlines = 0;
int curline = 0;

GC theGC; /* Context grafic pentru desen obisnuit */
GC xorGC; /* Context grafic pentru desenarea liniilor frante */

Cursor xhair_cursor;

/* Prototipuri de functii */

/* Acestea sunt apeluri inverse */
```

```
void start_rubberband(Widget w, XtPointer data, XEvent *p_event,
    Boolean *cdispatch);
void continue_rubberband(Widget w, XtPointer data, XEvent *p_event,
    Boolean *cdispatch);
void end_rubberband(Widget w, XtPointer data, XEvent *p_event,
    Boolean *cdispatch);

void handle_expose(Widget w, XtPointer client_data, XtPointer other);

/* Aceasta functie deseneaza liniile */
static void draw_line(Display *d, Window w, GC gc, int curline);

/*-----*/
void main(int argc, char **argv)
{
    Widget top_level, main_window, drawing_area;
    Arg args[MAXARGS];
    Cardinal argcount;
    int fg, bg;
    XGCValues xgcv;
    XtAppContext app;

    /* Creeaza primitiva shell de nivel superior si initializeaza kitul cu instrumente*/
    top_level = XtAppInitialize(&app, „Xm1ines“, NULL, 0,
        &argc, argv, NULL, NULL, 0);

    /* Apoi, primitiva ferestrei principale */
    argcount = 0;
    XtSetArg(args[argcount], XmNwidth, WIDTH); argcount++;
    XtSetArg(args[argcount], XmNheight, HEIGHT); argcount++;
    main_window = XmCreateMainWindow(top_level, „Main“,
        args, argcount);

    /* Creeaza suprafata de desenare */
    argcount = 0;
    XtSetArg(args[argcount], XmNresizePolicy, XmRESIZE_ANY);
    argcount++;
    drawing_area = XmCreateDrawingArea(main_window,
        „drawing_area“, args, argcount);
    XtManageChild(drawing_area);

    /* Ataseaza suprafata de desenare ferestrei principale */
    XmMainWindowSetAreas(main_window, NULL, NULL, NULL,
        NULL, drawing_area);

    /* Creeaza contextele grafice. Mai întâi obtine culorile pentru prim-plan si
    fundal din
    * resursele primitivei.
    */
    argcount = 0;
    XtSetArg(args[argcount], XmNforeground, &fg); argcount++;
    XtSetArg(args[argcount], XmNbackground, &bg); argcount++;
    XtGetValues(drawing_area, args, argcount);

    /* Defineste un context grafic cu aceste culori */
    xgcv.foreground = fg;
    xgcv.background = bg;
    theGC = XtGetGC(drawing_area, GCForeground | GCBackground,
        &xgcv);
```

```

/* Setează un context grafic cu modul exclusiv OR (pentru desenarea liniilor
frante)*/
xgcv. foreground = fg ^bg;
xgcv. background = bg;
xgcv.function = Gxor;
xorGC = XtGetGC(drawing_area, GCForeground |
    GCBackground | GCFunction, &xgcv);

/* Adauga apeluri inverse de tratare a evenimentelor de prezentare pentru
suprafata de desenare */
XtAddCallback(drawing_area, XmNexposeCallback, handle_expose,
    & drawing_area);

/* Creează un cursor subtire rectangular pentru suprafata de desenare */
xhair_cursor = XCreateFontCursor(XtDisplay(drawing_area),
    XC_crosshair);

/* Adauga subrutine de tratare a evenimentelor generate de butoane pentru
gestionarea desenului*/
XtAddEventHandler(drawing_area, ButtonPressMask, False,
    start_rubberband, NULL);
XtAddEventHandler(drawing_area, ButtonMotionMask, False,
    continue_rubberband, NULL);
XtAddEventHandler(drawing_area, ButtonReleaseMask, False,
    end_rubberband, NULL);

/* Realizează toate primitivele */
XtRealizeWidget(top_level);

/* Modifică cursorul pentru suprafata de desenare */
XDefineCursor(XtDisplay(drawing_area), XtWindow(drawing_area),
    xhair_cursor);

/* Creează un handle astfel încât cursorul se modifică în două fire subțiri
rectangulare și
* este restricționat la suprafata de desenare în timp ce butonul mouse-ului este
apasat.
* Acest lucru este realizat prin așa-numitul „grab”(apucare)
*/
XGrabButton(XtDisplay(drawing_area), AnyButton, AnyModifier,
    XtWindow(drawing_area), True, ButtonPressMask |
    ButtonMotionMask | ButtonReleaseMask, ,
    GrabModeAsync, GrabModeAsync,
    XtWindow(drawing_area), xhair_cursor);

/* Eliberează cursorul */
XFreeCursor(XtDisplay(drawing_area), xhair_cursor);

/* Porneste ciclul principal de tratare a evenimentelor */
XtAppMainLoop(app);
}
/*-----*/
/* start_rubberband
*
* Inceputul liniei frante
*/
void start_rubberband(Widget w, XtPointer data, XEvent *p_event,
    Boolean *cdispatch)
{
    int x = p_event->xbutton.x,

```

```

    y = p_event->xbutton.y;

/* Verificare sumara pentru a ne asigura ca nu depasim capacitatea sirului */
if(numlines > MAXLINES-1) numlines = MAXLINES-1;
curline = numlines;
numlines++;

lines[curline].x1 = x;
lines[curline].y1 = y;
lines[curline].x2 = x;
lines[curline].y2 = y;
draw_line(XtDisplay(w), XtWindow(w), xorGC, curline);
}
/*-----*/
/* continue_rubberband
*
* Gestioneaza miscarile mouse-ului în timpul desenarii unei linii frante
*/
void continue_rubberband(Widget w, XtPointer data, XEvent
    *p_event,
    Boolean *cdispatch)
{
    int x = p_event->xbutton.x,
    y = p_event->xbutton.y

/* Desenează o data la vechea locație (pentru a șterge linia) */
draw_line(XtDisplay(w), XtWindow(w), xorGC, curline);

/* Acum actualizează ultimul punct și redesenează linia */
lines[curline].x2 = x;
lines[curline].y2 = y;
draw_line(XtDisplay(w), XtWindow(w), xorGC, curline);
}
/*-----*/
/* end_rubberband
*
* Sfarsitul desenului format din linii frante
*/
void end_rubberband(Widget w, XtPointer data, XEvent
    *p_event,
    Boolean *cdispatch)
{
    int x = p_event->xbutton.x,
    y = p_event->xbutton.y;

/* Desenează o data la vechea locație (pentru a șterge linia) */
draw_line(XtDisplay(w), XtWindow(w), xorGC, curline);

/* Acum actualizează ultimul punct și redesenează linia în contextul grafic
normal */
lines[curline].x2 = x;
lines[curline].y2 = y;
draw_line(XtDisplay(w), XtWindow(w), theGC, curline);
}
/*-----*/
/* handle_expose
*
* Subrutina de tratare a evenimentelor de prezentare pentru suprafata de desenare
*/

```

```

void handle_expose(Widget w, XtPointer data, XtPointer other)
{
    XmDrawingAreaCallbackStruct *call_data =
        (XmDrawingAreaCallbackStruct*)other;
    XEvent *p_event = call_data->event;
    Window win = call_data->window;
    Display *p_display = XtDisplay(w);

    if (p_event->xexpose.count == 0)
    {
        int i;
        /* Sterge continutul ferestrei si deseneaza liniile în sirul „lines” */
        XClearWindow(p_display, win);

        if (numlines > 0)
        {
            XDrawSegments(p_display, win, theGC, lines, numlines);
        }

        /* Afiseaza mesajul (acesta este un exemplu de rezultat sub forma de text) */
        XDrawImageString(p_display, win, theGC, 50, 30, message,
            msglen);
    }
    /* ----- */
    /* draw_line
    *
    * Deseneaza o anumita linie din sirul de linii.
    */
    static void draw_line(Display *d, Window w, GC gc, int curline)
    {
        int x1 = lines[curline].x1, y1 = lines[curline].y1,
            x2 = lines[curline].x2, y2 = lines[curline].y2;

        XDrawLine(d, w, gc, x1, y1, x2, y2);
    }
}

```



Pentru a construi programul `xmlines` folosiți următoarea comandă:

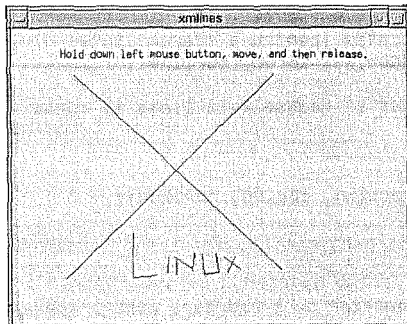
Sugestie

```
gcc -o xmlines xmlines.c -lXm -lXt -lX11 -lXext
```

Figura 23-2 prezintă rezultatul rulării programului `xmlines`.

Figura 23-2

Rezultatul rulării programului `xmlines`: afișarea câtorva linii.



Folosirea culorilor

X încorporează un model abstract de culori, ce conține multe din facilitățile componentelor hardware obișnuite pentru grafică color. Pentru a folosi culori într-o aplicație X, trebuie să înțelegeți întrebuințările modelului de culori X.

Moduri de vizualizare

X conține caracteristicile obișnuite ale componentelor hardware pentru afișare într-o structură de date numită `Visual`. X adaugă o caracteristică importantă pentru a încuraja partajarea hărților de culori, prin facilitarea accesului de citire sau citire-scriere pentru acestea. Celulele din hărțile de culori cu acces la citire-scriere pot fi modificate dinamic, în timp ce hărțile de culori read-only sunt fixe.

Membrul clasei structurii `Visual` indică facilitățile referitoare la culori ale afișării grafice de bază. Există șase clase distincte de `Visual`, clasificate în funcție de afișarea culorilor, a nuanțelor de gri precum și a noțiunii X de hărți de culori read-only și read-write, identificate după următoarele nume (definite în fișierul antet `<X11/X.h>`):

- Modul de vizualizare `DirectColor` reprezintă o afișare în care valoarea pixelului este descompusă în câmpuri de biți care se indexează în hărți de culori individuale pentru componentele R, G și B. Înregistrările din harta de culori pot fi schimbate dinamic. Această clasă de vizualizare este frecventă printre afișările cu planuri pe 24 de biți.
- Afișările modului de vizualizare `TrueColor` sunt identice cu cele ale modului `DirectColor`, dar hărțile lor de culori sunt fixe.
- Clasa de vizualizare `PseudoColor` modelează un tip obișnuit de componentă hardware de afișare: unul în care valoarea fiecărui pixel caută o valoare RGB, iar harta de culori poate fi modificată oricând.
- Afișările modului de vizualizare `StaticColor` sunt asemănătoare cu cele ale modului `PseudoColor`, cu deosebirea că harta de culori nu poate fi modificată.
- Modul de vizualizare `GrayScale` reprezintă un monitor cu nuanțe de gri ce permite modificarea hărții de intensitate. Modul de vizualizare `GrayScale` este gama de gri echivalentă modului de vizualizare `PseudoColor`.
- Modul de vizualizare `StaticGray` este asemănător cu `GrayScale`, dar are o hartă cu nivel de gri fixă. Afișările alb-negru (monocrome) sunt modelate de un mod de vizualizare `StaticGray` cu adâncimea 1.

Fiecare ecran dintr-o afișare X are cel puțin o structură de vizualizare asociată. Multe servere furnizează mai multe moduri de vizualizare pentru un ecran. Un server pentru o afișare pe 8 biți, de exemplu, poate furniza un mod de vizualizare `PseudoColor` de adâncime 8 și un mod de vizualizare `StaticGray` cu adâncimea 1. Astfel, ferestrele de pe același monitor pot fi folosite ca ferestre color sau, în cazul `StaticGray`, ca ferestre monocrome.



Observație

Chiar dacă un server poate utiliza mai multe moduri de vizualizare, unul dintre acestea este prestabilit. Se poate face referire la acest mod de vizualizare prestabilit folosind macro-ul `DefaultVisual(display, screen)`, care returnează un pointer către structura `Visual` a ecranului specificat din afișarea X identificată prin `display`.

Moduri de vizualizare disponibile

Pentru a vedea o listă a modurilor de vizualizare pe care serverul X al sistemului dumneavoastră le poate utiliza, porniți X (folosind comanda `startx`) și apoi tastați următoarea comandă la promptul interpretorului:

```
xdpinfo -display :0.0
```

Programul `xdpyinfo` afișează multe informații despre serverul X, inclusiv numărul de moduri de vizualizare. De obicei, serverul X poate utiliza șase moduri de vizualizare, cel prestabilit fiind un mod de vizualizare `PseudoColor` pe 8 biți.

De obicei, nu trebuie să selectați nici un mod de vizualizare, deoarece cel prestabilit este adecvat pentru cele mai multe aplicații. Dacă sistemul dumneavoastră are plăci grafice pe 24 de biți, iar modul de vizualizare prestabilit este pe 8 biți, veți dori probabil să selectați unul pe 24 de biți.



Secret

Pentru a utiliza un mod de vizualizare care nu este prestabilit într-o aplicație Motif, trebuie să specificați trei resurse – tipul de vizualizare, harta de culori și adâncimea – ale interpretorului de nivel superior înainte ca primitiva să fie sesizată.

Harta de culori X

În X, fiecare fereastră are asociată o hartă de culori care determină cum sunt transformate valorile pixelilor în culori (sau niveluri de gri, pentru monitoarele cu gamă de gri). Cu toate că componentele hardware permit o singură hartă de culori, X permite fiecărei ferestre să aibă propria hartă de culori, atât timp cât clasele de vizualizare ale ecranului nu sunt TrueColor, StaticColor sau StaticGray. Înainte ca valoarea pixelilor dintr-o fereastră să fie interpretată în conformitate cu o astfel de *hartă de culori virtuală*, harta de culori trebuie să fie instalată în harta de culori hardware. Aveți nevoie de un program de gestiune a ferestrelor pentru această operație dificilă. Prin convenție, cei mai mulți administratori de ferestre implementează o politică de instalare a hărții de culori a unei ferestre în componentele hardware imediat ce fereastra devine activă pentru introducerea de date.

● hartă de culori este o resursă a serverului X. În mod normal, nu trebuie să creați o nouă hartă de culori pentru aplicația dumneavoastră. Când serverul este pornit, el creează și instalează o hartă de culori prestabilită. În mod normal, serverul definește numai două celule color în aceasta hartă de culori prestabilită. Restul celulelor pot fi alocate și folosite de orice aplicație X.



Observație

Pentru a folosi culori dintr-o hartă de culori, trebuie să găsiți mai întâi identificatorul hărții, care este o variabilă de tipul ColorMap. Când creați o nouă hartă de culori, se returnează un identificator. Pentru harta de culori prestabilită, identificatorul este returnat de macro-ul DefaultColormap(display, screen), care vă solicită să identificați afișarea și ecranul. Puteți folosi DefaultScreen(display) ca argument pentru ecran în macro-ul DefaultColormap.

Culori dintr-o hartă de culori

Când folosiți culori într-o aplicație Motif, trebuie să specificați valoarea unui pixel. Valoarea pixelului implică o anumită culoare, în funcție de conținutul celei din harta de culori la care se referă. Primul pas în folosirea unei culori în X este de a obține indexul unei celule din harta de culori ce conține intensități de roșu, verde și albastru adecvate pentru culoarea pe care o doriți. Pentru a face acest lucru, solicitați serverului X să aloce culorii dumneavoastră o celulă dintr-o hartă de culori (specificată). Când solicitați o celulă a unei hărți de culori, trebuie să furnizați nivelurile pe care le doriți de roșu, verde și albastru; serverul X returnează un index pe care-l puteți folosi ca valoare a pixelului corespunzătoare acelei culori.



Observație

Deoarece multe aplicații pot folosi culori asemănătoare, X furnizează două modalități de alocare a celulelor:

- Celule partajate protejate la scriere (read-only)
- Celule private cu acces de tip citire-scriere (read-write)

Celule partajate protejate la scriere

Puteți alocă celule partajate protejate la scriere în orice clasă de vizualizare, dar pentru a alocă celule pentru citire-scriere, clasa de vizualizare trebuie să permită modificarea hărții de culori. Acest lucru înseamnă că clasa de vizualizare trebuie să fie DirectColor, PseudoColor sau GrayScale.

Pentru fiecare hartă de culori, serverul X ține evidența celulelor folosite curent. Când serverul primește o solicitare pentru o celulă partajată protejată la scriere într-o hartă de culori, el determină culoarea cea mai apropiată pe care componenta hardware o poate utiliza; apoi caută orice celulă de citire alocată anterior din acea hartă de culori, pentru a determina dacă nu cumva conține deja acea culoare. Dacă serverul găsește o astfel de celulă, returnează informații despre aceasta. Altfel, dacă clasa de vizualizare permite scrierea în harta de culori, serverul alocă o nouă celulă numai pentru citire, încarcă culoarea solicitată în celulă și returnează acea informație.



Observație

Ceea ce este interesant despre celulele hărților de culori partajate de tip read-only este că le puteți folosi în orice mod de vizualizare; pentru a alocă celule cu acces de tip citire-scriere ale hărților de culori private, trebuie să vă asigurați că modul de vizualizare este DirectColor, PseudoColor sau GrayScale.

Celule private cu acces la citire-scriere

Celulele private cu acces la citire-scriere au, de asemenea, unele avantaje. Aceste celule vă permit să modificați afișarea pixelilor cu diferite culori în orice moment. Această facilități poate fi folosită pentru afișarea imaginilor; puteți schimba culorile unei imagini schimbând înregistrările din harta de culori, fără să fie necesar să redeseñați imaginea cu noile valori ale pixelilor.

De asemenea, celulele hărților de culori private pot fi alocate într-un singur bloc contiguu astfel încât valorile pixelilor să fie succesive. Această facilități poate fi folosită dacă aplicația impune ca între culorile afișate și valorile pixelilor să existe o relație bine definită. Un exemplu îl reprezintă o imagine prin satelit afișată în nuanțe de gri care asociază un nivel de gri cu fiecare nivel de date din imaginea prin satelit.

Structura XColor

Când alocați culori sau definiți noi înregistrări în harta de culori, furnizați informații despre culoare într-o structură XColor, care este definită în fișierul antet <X11/Xlib.h> astfel:

```
typedef struct
{
    unsigned long pixel; /*Valoarea pixelului (dupa alocare) */
    unsigned short red, /* Intensitatea culorii roșu (0 - 65,535) */
                  green, /* Intensitatea culorii verde (0 - 65,535) */
                  blue; /* Intensitatea culorii albastru (0 - 65,535) */
    char flags; /* Utilizat la stocarea culorilor */
    char pad; /* Adaugata pentru a echilibra mărimea structurii */
};
```



Observație

Ar trebui să specificați o culoare indicând intensitățile componentelor RGB în câmpurile *red*, *green* și *blue*. Aceste intensități pot lua valori între 0 și 65535; 0 înseamnă intensitate nulă, iar 65535 înseamnă intensitate maximă. Serverul scalează automat aceste valori la domeniul de intensități cerut de componenta hardware a ecranului de afișare.

Câmpul denumit pixel este valoarea pixelului folosită pentru a afișa culoarea corespunzătoare valorii RGB în roșu, verde și albastru. Când se alocă o culoare, trebuie furnizate valorile RGB ale acesteia; serverul X returnează valoarea pixelului corespunzătoare celei din harta de culori care are acea culoare.

Alocarea celulelor cu acces read-only ale hărților de culori

Biblioteca Xlib furnizează două funcții pentru alocarea celulelor partajate cu acces read-only ale unei hărți de culori:

- XAllocColor
- XAllocNamedColor

Funcția XAllocColor vă solicită să definiți o structură XColor și apoi să completați câmpurile roșu, verde și albastru cu nivelurile RGB ale culorii pe care o doriți. Trebuie să specificați, de asemenea, harta în care doriți să alocați culoarea. De obicei, folosiți funcția XAllocColor în conjuncție cu XparseColor, care acceptă numele unei culori și setează valorile corespunzătoare RGB într-o structură XColor.

Pentru a aloca o culoare cyan deschis (light cyan) în harta de culori prestabilită și pentru a o folosi ca fundal al unei ferestre, de exemplu, puteți folosi următorul cod (vezi capitolul 5 pentru informații despre denumirea culorilor în aplicații X):

```
Display *disp;
XColor color;
Colormap default_cmap;
unsigned long bgpixel;

default_cmap = DefaultColormap(disp,
                               DefaultScreen(disp));

/* Incercati sa alocati o celula „cyan deschis” în harta de culori */

if (XParseColor(disp, default_cmap, „light cyan”,
               &color) == 0 ||
    XAllocColor(disp, default_cmap, &color) == 0)
{
    /* Folositi ca fundal culoarea alb în caz de esec */
    bgpixel = WhitePixel(disp, DefaultScreen(disp));
}
else
{
    /* Celula hartii de culori a fost alocata cu succes */
    bgpixel = color.pixel;
}

/* Folositi „bgpixel” ca fundal cand creati fereastra */
```

Mai întâi, funcția XParseColor setează valorile RGB pentru culoarea denumită *cyan deschis*. Apoi, funcția XAllocColor solicită o celulă color read-only care are valoarea RGB stabilită anterior. Dacă totul merge bine, ambele funcții returnează valori diferite de zero și membrul pixel al variabilei color conține valoarea pixelului care trebuie folosită pentru culoarea *cyan deschis*. Dacă funcțiile eșuează, trebuie să includeți câteva mijloace pentru a ține situația sub control. În acest exemplu, dacă alocarea eșuează, culoarea alb este folosită pentru fundal. Serverul alocă întotdeauna culorile alb și negru pentru harta de culori prestabilită. Vă puteți referi la aceste culori folosind macro-urile WhitePixel(display, screen) și respectiv BlackPixel(display, screen).

Serverul X furnizează o bază de date ce conține culori pe care aplicațiile le pot folosi pentru a converti numele textuale ale culorilor în intensități de roșu, verde și albastru, adecvate pentru ecranul de afișare respectiv. Funcțiile XParseColor și XAllocNamedColor folosesc această bază de date.

Funcția XAllocNamedColor este asemănătoare cu XAllocColor, cu excepția faptului că preia direct numele unei culori ca șir. Când folosiți funcția XAllocNamedColor, exemplul anterior pentru alocarea unei celule cyan deschis se modifică astfel:

```
Xcolor exact; /* Definitia RGB exacta a culorii */

if (XAllocNamedColor(disp, default_cmap, „light cyan”,
                    &exact, &color) == 0)
{
    /* Folositi un fundal alb în caz de esec */
    bgpixel = WhitePixel(disp, DefaultScreen(disp));
}
else
{
    /* Celula hartii de culori a fost alocata cu succes */
    bgpixel = color.pixel;
}
```

Funcția XAllocNamedColor necesită două argumente XColor. În structura XColor denumită exact, funcția XAllocNamedColor returnează valoarea exactă RGB pentru culoarea solicitată (din baza de date). În structura color, ea returnează cea mai apropiată culoare pe care componentele hardware le pot utiliza și valoarea pixelului corespunzătoare celulei alocate din harta de culori (când alocarea reușește).

Alocarea celulelor cu acces read-write din harta de culori

Unele aplicații necesită ca în harta de culori culorile să fie alocate într-un anumit mod. Puteți, de exemplu, să afișați un vector bidimensional de date meteo transmise prin satelit ca o imagine în care culoarea fiecărui pixel reprezintă o informație. Când afișați o astfel de imagine, doriți probabil ca toate valorile pixelilor să apară în ordine, astfel încât să puteți transpune cu ușurință datele în culori. Puteți dori, de asemenea, să modificați transpunerea datelor în culori pentru a obține caracteristici ale datelor. Această problemă se rezolvă cel mai bine prin alocarea unui bloc contiguu de celule private cu aceea read-write din harta de culori.

Pentru a aloca celule color neprotejate la scriere, apelați funcția XAllocColorCells. Înainte de a folosi această funcție, desigur, trebuie să vă asigurați că clasa de vizualizare a ecranului permite modificări în harta de culori (clasa trebuie să fie DirectColor, PseudoColor sau GrayScale).

Iată cum ar trebui să apelați funcția XAllocColorCells:

```
Display *disp;
Colormap cmap;
Bool contig;
unsigned long planes [ ];
unsigned int nplanes;
unsigned long pixels [ ];
unsigned int npixels;

/* Harta de culori in care sunt alocate celulele */
/* True = aloca planuri contigue */
/* Sir ce va depozita masca planurilor */
/* Numarul planurilor ce vor fi alocate */
/* Sir ce va depozita valorile pixelilor */
/* Numarul pixelilor ce vor fi alocati */

if (!XAllocColorCells(disp, cmap, contig, planes, nplanes, pixels,
                    npixels) )
{
    /* Eroare la alocarea celulelor color */
    /* ... */
}
```

X vă solicită să specificați celulele cu acces read-write din harta de culori într-o manieră unică. Trebuie să specificați numărul valorilor pixelilor (npixels) și numărul planurilor (nplanes); npixels trebuie să fie pozitiv, iar nplanes poate fi pozitiv sau zero. În schimb, serverul rezervă

`npixels*2nplanes` celule din harta de culori și returnează informațiile despre celulele ce pot fi folosite în vectorii numiți `pixels` și `planes`. `npixels` valori sunt returnate în `pixels` și `nplanes` maști de biți sunt returnate în vectorii `planes`.

Dacă doriți să alocați o singură celulă cu acces read-write din harta de culori, puteți apela funcția `XAllocColorCells` astfel:

```
Display *disp;
Colormap cmap; /* Harta de culori în care sunt alocate celule */
unsigned long pixels [1];

if (!XAllocColorCells (disp, cmap, False, NULL, 0, pixels, 1) )
{
    /* Eroare la alocarea celulelor color */
    /* ... */
}
/* Celula color alocata cu succes. Valoarea pixelului este pixels[0] */
```

După ce apelați funcția `XAllocColorCells` pentru a alocă celule private din harta de culori neprotejate la scriere, trebuie să stocați culorile în aceste celule înainte de a le folosi. Folosiți funcția `XStoreColor` pentru a modifica valoarea RGB corespunzătoare unei singure valori de pixel; pentru a face acest lucru trebuie să folosiți o structură `XColor`. Setări câmpurile `red`, `green` și `blue` pe nivelurile dorite de culoare primară. Setări câmpul `pixel` pe valoarea pixelului unei celule alocate cu acces read-write. Apoi apelați funcția `XStoreColor` astfel:

```
Colormap colormap;
XColor color;
/* Considerati ca celula color este o celula neprotejata la scriere alocata
anterior.
* Urmatorul cod stocheaza culoarea „red” în aceasta celula.
*/
color.pixel = colorcell;
color.red = 65535;
color.green = 0;
color.blue = 0;
color.flags = DoRed | DoGreen | DoBlue;
XStoreColor (disp, colormap, &color);
```

Câmpul `flags` din variabila `color` indică nivelurile culorii primare din celula hărții de culori care trebuie actualizate.



Sugestie

Pentru a seta o singură celulă din harta de culori pe o culoare identificată după nume și nu după componentele RGB, puteți folosi funcția `XStoreNamedColor`. Pentru a seta valoarea pixelului `colorcell` pe roșu, folosiți următoarele:

```
XStoreNamedColor (p_disp, colormap, „red”, colorcell,
DoRed | DoGreen | DoBlue);
```

Dacă aveți de inițializat câteva celule cu acces read-write din harta de culori, puteți folosi funcția `XStoreColors` pentru a stoca în același timp culorile în ele. Așa cum este de așteptat, această funcție preia un vector de structuri `XColor` și numărul acestora ca argumente, astfel:

```
XColor colors [ ]; /* Sir de culori */
int ncolors; /* Numarul campurilor din sirul de culori */
.
.
/* Seteaza sirul „colors” inaintea apelarii functiei XStoreColors */
XStoreColors(p_disp, colormap, colors, ncolors);
```

Eliberarea culorilor

Dacă folosiți celule private sau partajate ale hărții de culori, ar trebui să eliberați culorile atunci când nu mai aveți nevoie de ele. Când o aplicație se încheie, serverul X eliberează automat culorile folosite de acea aplicație. Dacă aplicația dumneavoastră alocă deseori culori (de exemplu, poate alocă culori de fiecare dată când o imagine este afișată într-o fereastră), ar trebui să apelați funcția `XFreeColors` pentru a le elibera pe cele de care nu mai aveți nevoie, astfel:

```
Display *disp;
Colormap cmap; /* Celule eliberate în aceasta harta de culori */
unsigned long pixels [ ]; /* Identifica celulele ce sunt eliberate */
int numpix; /* Numarul celulelor eliberate */
unsigned long planes; /* Identifica planurile eliberate */
```

```
XFreeColors (disp, colormap, pixels, numpix, planes);
```

`XFreeColors` eliberează culorile alocate de oricare dintre funcțiile următoare: `XAllocColor`, `XAllocNamedColor`, `XAllocColorCells` și `XAllocColorPlanes`. Când eliberați una sau mai multe celule cu acces read-only alocate de `XAllocColor` sau `XAllocNamedColor`, furnizați vectorul cu valorile pixelilor, numărul acestora și un zero pentru argumentul `planes`.

Pentru celule private neprotejate la scriere, folosiți un OR logic pentru măștile planului care au fost returnate de un apel anterior al funcției `XAllocColorCells`.

Afișarea unei imagini

Serverul X include suport pentru imagini sub forma hărților de pixeli și hărților de biți, în timp ce Xlib suportă structura `XImage`, care vă permite să manipulați imaginile local, pe sistemul în care rulează aplicația dumneavoastră, nu pe server.

Crearea unei hărți de pixeli

O hartă de pixeli este o desenabilă, ceea ce înseamnă că puteți desena într-o bucată de pixeli ca într-o fereastră. La fel ca o fereastră, o hartă de pixeli poate fi considerată ca un masiv rectangular de pixeli (un rastru), cu fiecare locație capabilă de a înmagazina valoarea unui pixel. Puteți, de asemenea, să imaginați o hartă de pixeli sub forma mai multor planuri de biți rectangulari, ce conține atâtea planuri câți biți se găsesc în valoarea pixelului.

Pentru ferestre, valorile pixelului în rastru sunt afișate constant. Componentele hardware citesc valorile pixelului și le transpun în culori sau în niveluri de gri, în funcție de facilitățile monitorului.

Spre deosebire de ferestre, conținutul unei hărți de pixeli nu este vizibil până când nu este copiat într-o fereastră. Astfel, puteți considera o hartă de pixeli ca o suprafață de desenare în afara ecranului — o suprafață de memorie în care puteți salva un desen sau o imagine. De fapt, puteți pregăti desene într-o hartă de pixeli și le puteți afișa atunci când este necesar folosind funcția `XCopyArea`.

Hărțile de pixeli sunt folosite, în primul rând, pentru a desena imagini și a stoca modele pentru acoperirea suprafețelor (tiling). Hărțile de pixeli folosite pentru acoperire de suprafețe sunt mici (de obicei, nu mai mult de 32x32 pixeli), dar cele folosite pentru stocarea imaginilor pot fi destul de mari.

Ca și ferestrele, hărțile de pixeli sunt resurse întreținute de serverul X. Înainte de a folosi o hartă de pixeli, trebuie să o creați apelând funcția `XCreatePixmap` din biblioteca Xlib. Această funcție returnează un identificator de resursă de tipul `Pixmap`, folosit la referirile la harta de pixeli în solicitările de desenare.

Să presupunem că doriți să creați o hartă de pixeli pentru a desena câteva cifre ce nu apar pe ecran. Puteți crea o hartă de pixeli cu următorul cod:

```
Display *disp; /* Identifica afisarea X */
Window root_win; /* Identificatorul ferestrei root (radacina) */
Pixmap pmap1; /* Harta de pixeli ce este creata */
unsigned int width, /* Latimea hartii de pixeli (exprimata in pixeli) */
            Height, /* Inaltimea hartii de pixeli (in pixeli) */
            Depth; /* Numarul de biti folositi pentru a indica valoarea
pixelului */
.
/* Creeaza o harta de pixeli */
width = 100;
height = 50;
depth = DefaultDepth(disp, DefaultScreen(disp));
root_win = RootWindow(disp, DefaultScreen(disp));
pmap1 = XCreatePixmap (disp, root_win, width, height, depth);
```

Când creați o hartă de pixeli, specificați dimensiunile (width=lățimea și height=înălțimea) și adâncimea ei (numărul de biti în valori pixel pe care harta de biti trebuie să-i stocheze).

Al doilea argument al funcției `XCreatePixmap` trebuie să fie un identificator al unei desenabile create anterior. Serverul X folosește acest argument pentru a determina ecranul pentru care creați harta de pixeli. Puteți folosi identificatorul oricărei ferestre sau hărți de biti valide. O soluție simplă este să folosiți identificatorul ferestrei rădăcină.



Atenție!

În X, fiecare fereastră și hartă de pixeli este creată pentru un anumit ecran. Într-o afișare X cu ecrane multiple, nu puteți copia o hartă de pixeli dintr-un ecran într-o fereastră a altui ecran, chiar dacă ambele au aceeași adâncime.

Desenarea într-o hartă de pixeli

Când o hartă de pixeli validă este disponibilă, puteți desena în ea exact ca într-o fereastră. Iată cum puteți desena câteva cifre într-o hartă de pixeli:

```
/* Desenarea într-o harta de pixeli (considerati ca contextul grafic este creat )
*/
XFillRectangle(disp, pmap1, theGC, 10, 10, 20, 20);
XFillArc (disp, pmap1, theGC, 30, 30, 20, 20, 0, 360*64);
```

La fel ca în cazul ferestrelor, trebuie să furnizați un context grafic (GC) cu toate atributele grafice, precum culorile pentru fundal și prim-plan. În cazul hărților de pixeli, folosiți același sistem de coordonate ca în cazul ferestrelor – cu originea în colțul din stânga sus, sensul pozitiv al axei X orientat spre dreapta iar cel al axei Y orientat în jos.



Secret

Există totuși câteva diferențe minore între desenarea în harta de pixeli și cea în ferestre. Ferestrele sunt întotdeauna afișate pe ecran, în timp ce hărțile de pixeli nu sunt niciodată afișate. Una din diferențele principale este că ferestrele au o culoare de fundal asociată, în timp ce hărțile de pixeli nu. Astfel, nu puteți umple harta de pixeli folosind culoarea de fundal apelând funcția `XClearArea`. În schimb, trebuie să folosiți `XFillRectangle` pentru a umple tot pixelii din harta de pixeli folosind culoarea de fundal curentă. Când creați o harta de pixeli, conținutul ei este nedefinit, astfel că întotdeauna trebuie s-o umpleți cu o valoare cunoscută, ca de exemplu culoarea de fundal.

Hărțile de pixeli nu generează decât evenimentele `GraphicsExpose` și `NoExpose`, ce au loc în timpul procesului de copiere între ferestre și hărți de pixeli.

Afișarea unei hărți de pixeli

După ce ați creat o hartă de pixeli și ați desenat în interiorul acesteia, o puteți afișa folosind funcția `XCopyArea`. Pentru a afișa harta de pixeli `pmap1` în fereastra `win` la coordonatele `x`, `y` de exemplu, folosiți următorul cod:

```
/* Copierea desenului din harta de pixeli în fereastra */
XcopyArea(p_disp, pmap1, win, theGC, 0, 0, width, height, x, y);
```

Eliberarea hărților de pixeli

Deoarece cele mai multe afișări X au memorie limitată în afara ecranului, ar trebui să eliberați resursele hărților de pixeli imediat ce ați terminat utilizarea acestora. Puteți elibera o hartă de pixeli apelând funcția `XFreePixmap` astfel:

```
Pixmap pmap1
```

```
/* Creeaza harta de pixeli „pmap1” si o foloseste */
/* ... */
XFreePixmap (p_disp, pmap1);
```

Toate resursele (inclusiv hărțile de pixeli) folosite de aplicația dumneavoastră sunt eliberate automat când aplicația dumneavoastră se încheie. Trebuie să eliberați explicit hărțile de pixeli numai dacă folosiți numeroase hărți de pixeli mari.

Folosirea hărților de biti

O *hartă de biti* este o hartă de pixeli de adâncime 1. Puteți crea o hartă de biti specificând adâncimea 1 când apelați funcția `XCreatePixmap`. Deoarece există numai un bit în fiecare locație a hărții de biti, o hartă de biti este un model format din cifrele unu și zero.

Hărțile de biti se folosesc oarecum diferit față de hărțile de pixeli. Deoarece hărțile de pixeli stochează întreaga valoare a pixelului, ele sunt copiate de obicei direct în ferestre și afișate. Hărțile de biti, totuși, sunt în cele mai multe cazuri folosite ca șabloane (sau matrite), prin intermediul cărora valorile pixelilor de prim-plan și fundal se aplică rastrului unei suprafețe de desenare. Mai întâi, harta de biti este așezată peste rastrul suprafeței de desenare (repetând harta de biti, dacă este necesar, pentru a acoperi întregul rastru). Apoi, culoarea prim-planului se aplică tuturor pixelilor unde harta de biti (șablonul) are cifra unu.

Altă utilizare a hărților de biti este ca mască de decupare. Operația este asemănătoare cu folosirea hărții de biti ca șablon, cu excepția faptului că operațiile grafice sunt executate numai pentru acei pixeli din rastru care au cifra unu în modelul hărții de biti folosit ca șablon.

Datorită utilizărilor speciale ale hărților de biti, ele sunt folosite pe scară largă chiar și atunci când afișarea suportă mai multe planuri.

Cu toate că hărțile de biti sunt folosite deseori ca matrite și măști de decupare, puteți afișa o hartă de biti direct într-o fereastră sau s-o copiați într-o hartă de pixeli. Deoarece harta de biti este de adâncime 1, iar desenabila (harta de pixeli sau fereastra) poate avea o adâncime mai mare decât 1, nu puteți folosi funcția `XCopyArea` pentru această operație. Biblioteca Xlib furnizează altă funcție, `XCopyPlane`, special în acest scop.

`XCopyPlane` vă solicită să specificați o suprafață rectangulară într-o hartă de biti sursă și o desenabilă de destinație. Serverul copiază harta de biti în harta de pixeli destinație astfel: folosește harta de biti ca șablon și, utilizând culoarea prim-planului (specificată într-un context grafic), desenează acei pixeli în suprafața de desenare unde harta de biti are cifra unu.

Puteți identifica harta de biți ca fiind unul din planurile dintr-o hartă arbitrară de pixeli. Pentru a face acest lucru, specificați o mască cu exact un bit setat pe 1. Acea mască identifică un plan de biți în harta de pixeli sursă. Acest plan de biți este folosit ca șablon pentru operația de copiere.

În continuare sunt prezentate câteva linii de cod care copiază harta de biți citită de `XreadBitmapFile` într-o fereastră, folosind culoarea curentă a prim-planului dintr-un context grafic:

```
Display disp;           /* Identifica afisarea X */
Window win;            /* Fereastra de destinatie */
GC theGC;              /* Contextul grafic folosit pentru copiere */
Pixmap my_bmp;         /* Harta de biti creata de functie */
int xh_bmp, yh_bmp;     /* Coordonatele punctului fierbinte */
unsigned int w_bmp, h_bmp; /* Latimea si inaltimea hartii de biti */
unsigned long planemask; /* Identifica harta de biti sursa */
int xsrc, ysrc;         /* Coltul dreptunghiului sursa */
int xdest, ydest;       /* Copiaza in acest punct */

/* Citirea hartii de biti folosind „XreadBitmapFile“. Considerati ca datele
 * dintr-o harta de biti sunt localizate într-un fisier denumit „testicon.xbm“
 */
root_win = RootWindow(disp, DefaultScreen(disp));

if (XreadBitmapFile (disp, root_win, „testicon.xbm“,
                    &w_bmp, &h_bmp, &my_bmp,
                    &xh_bmp, &yh_bmp) != BitmapSuccess)
{
    fprintf (stderr, „Failed to read bitmap file!\n”);
    /* Iesiti daca nu puteti continua */
}

/* Harta de biti „my_bmp“ este gata pentru a fi folosita */

xsrc = 0;
ysrc = 0;
xdest = 10;
ydest = 10;
planemask = 1;

XCopyPlane (disp, my_bmp, win, theGC, xsrc, ysrc,
            w_bmp, h_bmp, xdest, ydest, planemask);
```

Remarcați că masca ce identifică harta de biți din `my_bmp` este setată pe 1, deoarece `my_bmp` este deja o hartă de biți și, ca atare, are numai un plan de biți.

Rezumat

În prezent, utilizatorii se așteaptă ca produsele software să conțină o interfață grafică ușor de folosit. În Linux, interfața grafică este sistemul X Window System – standardul pentru toate stațiile de lucru UNIX. Sistemul X furnizează facilitățile de bază în ceea ce privește lucrul cu ferestre și afișarea grafică, fără a depăși prea mult acest cadru. Facilitățile de grafică cele mai interesante sunt furnizate de seturi de instrumente cum ar fi Motif. Motif este standardul de facto al interfeței grafice cu utilizatorul pentru stațiile de lucru UNIX. Cu toate că Linux nu este distribuit cu Motif, puteți cumpăra această aplicație cu numai 100 \$. Deoarece Motif este cea mai

utilizată interfață grafică de utilizator, acest capitol vă furnizează o introducere în programarea Motif. Din lectura acestui capitol, ați învățat următoarele:

- ▶ Xlib este biblioteca de funcții C ce oferă acces la facilitățile de bază ale sistemului X Window System. Chiar cu ajutorul bibliotecii Xlib, este plictisitor să creați elemente simple de interfață cu utilizatorul, cum ar fi butoanele și meniurile. Aveți nevoie de un set de instrumente de nivel mai avansat, cum ar fi Motif, pentru a construi cu ușurință interfețe grafice cu utilizatorul. (Unii sunt de părere că aveți nevoie de un nivel de abstractizare mai avansat decât cel furnizat de Motif.)
- ▶ Motif se bazează pe X Toolkit Intrinsics sau Xt Intrinsics, care este un set de funcții utile pentru administrarea primitivelor – obiecte grafice pentru interfața cu utilizatorul. Xt Intrinsics oferă un set de bază de primitive pe care sunt construite alte seturi de primitive (precum Motif).
- ▶ Programele Motif se bazează pe primitive pentru a implementa interfața cu utilizatorul. Caracteristicile primitivelor sunt controlate prin setarea variabilelor cunoscute ca *resurse*.
- ▶ Un program Motif tipic creează câteva primitive, le setează resursele și apoi execută un ciclu care procesează evenimente (cum ar fi intrări de la mouse și tastatură sau solicitarea de a redesena conținutul unei ferestre). Acest stil de programare este cunoscut ca *programare condusă prin evenimente*.
- ▶ Primitivele Motif folosesc o arhitectură orientată pe obiecte (chiar dacă acestea au fost implementate în C) care moștenește caracteristicile unui set de primitive de bază. Cele trei tipuri de primitive de bază sunt shell, autonome și manager.
- ▶ Cele mai multe distribuții Motif includ un program de exemplificare numit *periodic*. Puteți vedea primitivele rulând acest program demonstrativ.
- ▶ Cu toate că primitivele Motif simplifică procesul de construire a elementelor de interfață cu utilizatorul, cum ar fi butoanele, barele de meniuri și casetele listă, aveți nevoie de funcțiile de bază din biblioteca Xlib pentru afișarea textului, graficii și imaginilor.
- ▶ Contextul grafic (GC) este fundamentul oricărei afișări grafice ce folosește sistemul X Window System. Prin intermediul contextului grafic puteți specifica toate atributele grafice, cum ar fi culoarea și fontul.
- ▶ Hărțile de pixeli sunt blocuri de memorie în afara ecranului, în care puteți desena grafică și text exact ca într-o fereastră de pe monitor. Serverul X administrează hărțile de biți.
- ▶ Înainte de a apela funcțiile de desenare Xlib în programele Motif, trebuie să obțineți un context grafic.

Capitolul 24

Procesarea textului în Linux

În acest capitol

- ▶ Veți edita text în Linux
- ▶ Veți folosi editorul orientat spre linie `ed`
- ▶ Veți folosi editorul de text pe tot ecranul `vi`
- ▶ Veți edita fișierele text cu GNU Emacs
- ▶ Veți obține asistență online în GNU Emacs
- ▶ Veți înțelege formatul unei pagini `man`
- ▶ Veți pregăti o pagină `man` cu `groff`

Procesarea textelor se referă la toate aspectele creării, editării și formătărilor documentelor ce conțin text. Forma cea mai simplă de procesare a textului este pregătirea unui fișier text simplu, lucru pe care va trebui să-l faceți deseori, deoarece majoritatea fișierelor de configurare din Linux sunt fișiere text simplu. În acest scop, Linux oferă o gamă de editoare, de la editorul standard din UNIX `vi` la atotputernicul GNU Emacs.

Pentru a pregăti text formatat în Linux, trebuie să folosiți un limbaj de marcare precum `groff`. Cu un astfel de limbaj, inserați comenzi speciale de formatare într-un fișier text simplu, iar un program de formatare procesează fișierul text pentru a genera documentul formatat în vederea vizualizării sau tipăririi. Este posibil să fiți deja familiarizat cu un limbaj de marcare mai recent, numit Hypertext Markup Language (HTML), care este folosit ca format standard pentru documente în World Wide Web.



Observație

Chiar dacă folosiți o aplicație „what you see is what you get – ceea ce vezi este ceea ce vei obține” (cunoscută și sub forma abreviată WYSIWYG) în Microsoft Windows sau Macintosh pentru a pregăti textul formatat, trebuie să învățați elementele de bază ale unui limbaj de marcare dacă doriți să pregătiți o pagină `man` – asistență text online, disponibilă prin comanda `man`.

Acest capitol descrie facilitățile de procesare ale textului în Linux. Capitolul începe cu editoarele de text `ed`, `vi` și GNU Emacs. A doua jumătate a capitolului descrie cum să folosiți programul de formatare a textului `groff` pentru a pregăti o pagină `man`.

Editarea textului cu `ed` și `vi`

Editarea textului este o parte importantă a tuturor sistemelor de operare, inclusiv Linux. În Linux, trebuie să creați și să editați o varietate de fișiere text:

- Fișiere de configurare a sistemului, inclusiv `/etc/fstab`, `/etc/hosts`, `/etc/inittab`, `/etc/X11/XF86Config` și multe altele
- Fișiere utilizator precum `.newsrsc` și `bash_profile`
- Mesaje de poștă și articole cu mesaje

- Fișiere script ale interpretorului de comenzi
- Fișiere script Perl și Tcl/Tk
- Programe C sau C++

Toate sistemele UNIX, inclusiv Linux, sunt livrate cu două editoare de text:

- `ed`, un editor de text orientat spre linie
- `vi`, un editor de text pe tot ecranul, ce poate utiliza setul de comenzi al unui editor anterior numit `ex`

În Linux, `vi` și `ex` sunt emulate de alt editor de text numit `vim`, pe care-l puteți invoca folosind comanda `vi`.



Sugestie

Cu toate că `ed` și `vi` sunt mai complicate decât alte editoare de text cu interfață grafică, ar trebui să învățați comenzile de bază ale acestor două editoare, pentru că există situații când acestea sunt singurele disponibile. Când o să aveți probleme cu sistemul și Linux refuză să pornească de pe hard-disc, de exemplu, va trebui să încărcați sistemul de operare de pe o dischetă. În acest caz, trebuie să editați fișierele sistem cu editorul `ed`, deoarece acest editor este suficient de mic pentru a încăpea pe o dischetă.

Așa cum veți vedea în secțiunile următoare, comenzile de editare de bază ale programelor `ed` și `vi` nu sunt dificil de învățat.

Folosirea editorului `ed`

Editorul de text `ed` lucrează cu un *buffer* – o zonă de stocare în memorie unde se află textul până la stocarea explicită a acestuia într-un fișier. Trebuie să folosiți acest editor numai când porniți o versiune minimală a sistemului Linux (de exemplu, de pe o dischetă de boot), iar sistemul nu poate utiliza modul de editare pe tot ecranul.

Invocarea editorului `ed`

Pentru a invoca `ed` folosiți următoarea sintaxă:

```
ed [-] [-G] [-s] [-pprompt-string] [filename]
```

Argumentele aflate între paranteze pătrate sunt opționale. Următoarea listă explicitează argumentele:

- `-` cratima suprimă tipărirea numărului de caractere și a mesajelor de diagnostic
- `-G` forțează compatibilitatea cu versiunile anterioare
- `-s` are aceeași semnificație ca și cratima
- `-p prompt-string` setează șirul (string) afișat ca prompt (șirul prestabilit este `vid`)
- `filename` este numele fișierului ce va fi editat

Învățarea editorului `ed`

Când folosiți editorul `ed`, lucrați într-unul din următoarele moduri:

- *Command mode* (modul comandă) este modul prestabilit. În acest mod, orice tastați este interpretat ca o comandă. După cum puteți vedea în secțiunea „Rezumatul comenzilor `ed`”, editorul dispune de un set de comenzi simplu, în care fiecare comandă constă dintr-un singur caracter.
- *Text-input mode* (modul introducere text) vă permite să introduceți text într-un buffer. Puteți comuta pe modul introducere text folosind comenzile `a` (`append-adăugare`), `c`

(change-modificare) sau **i** (insert-inserare). După introducerea liniilor de text puteți părăsi acest mod tastând un punct (.) pe o linie separată.

Editorul **ed** lucrează cu noțiunea de *linie curentă* (current line) – linia la care se aplică comenzile pe care le tasteți. Fiecare linie are o adresă: numărul acesteia. Puteți aplica o comandă unui grup de linii prefixând comanda cu adresa grupului. Comanda **p**, de exemplu, afișează pe ecran linia curentă. Pentru a vedea primele zece linii, folosiți următoarea comandă:

```
1,10p
```

Într-o comandă, punctul (.) se referă la linia curentă, iar semnul dolar (\$) se referă la ultima linie. Astfel, următoarea comandă șterge toate liniile, începând de la linia curentă până la ultima:

```
.,$D
```

Examinarea unei sesiuni demonstrative cu ed

Următorul exemplu vă arată cum puteți începe editarea unui fișier în **ed**:

```
ed -p: /etc/fstab
532
:
```

Acest exemplu folosește opțiunea **-p** pentru a seta promptul la caracterul două puncte (:) și deschide fișierul **/etc/fstab** pentru editare. Este utilă introducerea unui caracter drept prompt, deoarece, în absența unui prompt, distincția între cele două moduri în care se poate găsi editorul **ed** se face cu mare dificultate.

Editorul **ed** deschide un fișier, raportează numărul caracterelor acestuia (532), afișează prompt-ul (:) și așteaptă o comandă.

După ce **ed** deschide un fișier pentru editare, linia curentă este ultima linie a fișierului. Pentru a vedea numărul liniei curente folosiți comanda **.=** astfel:

```
: .=
7
```

Acest rezultat indică faptul că fișierul **/etc/fstab** are șapte linii. (Fișierul **/etc/fstab** din sistemul dumneavoastră poate avea un număr diferit de linii.) Următorul exemplu vă arată cum puteți vedea toate aceste linii:

```
:1,$p
/Dev/hda1      /          ext2      defaults  1 1
/Dev/sda1      /usr/local ext2      defaults  1 2
/Dev/hda2      swap       swap      defaults  0 0
/Dev/sda2      swap       swap      defaults  0 0
/Dev/fd0       /mnt/floppy ext2      noauto    0 0
/Dev/cdrom     /mnt/cdrom iso9660   noauto,ro 0 0
None          /proc      proc      defaults  0 0
:
```

Pentru a vă deplasa la o anumită linie, tasteți numărul ei astfel:

```
:1
/Dev/hda1      ext2      defaults  1 1
:
```

Editorul răspunde afișând linia respectivă.

Să presupunem că doriți să ștergeți linia ce conține șirul de caractere **dosc**. Pentru a căuta un șir, tasteți un slash (/), urmat de șirul pe care vreți să-l localizați, astfel:

```
:/cdrom
/Dev/cdrom     /mnt/cdrom iso9660   noauto,ro 0 0
:
```

Editorul localizează linia ce conține șirul și o afișează. Linia respectivă devine linie curentă. Pentru a șterge linia, folosiți comanda **d** astfel:

```
:d
:
```

Pentru a înlocui un șir cu altul, folosiți comanda **s**. Pentru a înlocui **cdrom** cu șirul **cd**, de exemplu, folosiți următoarea comandă:

```
:s/cdrom/cd/
/Dev/cd          /mnt/cdrom      iso9660        noauto,ro      0 0
:
```

Pentru a insera o linie în fața liniei curente, folosiți comanda **i** astfel:

```
:i
    (tastati linia pe care vreți să o inserati)
.    (tastati un singur punct)
:
```

Puteți introduce câte linii doriți. După ultima linie introduceți un punct (.) pe o linie separată. Acel punct marchează sfârșitul modului de introducere text și editorul este comutat pe modul comandă. În acest caz, puteți observa că **ed** a comutat pe modul comandă, deoarece vedeți promptul (:) .



Sugestie

```
:wq
535
```

Dacă sunteți satisfăcuți de modificările efectuate, le puteți scrie în fișier folosind comanda **w**. Dacă doriți să salvați modificările și să părăsiți editorul, puteți tasta pur și simplu **wq** pentru a realiza amândouă operațiile simultan, astfel:

Editorul **ed** salvează modificările efectuate în fișier, afișează numărul caracterelor pe care le-a salvat și își încheie execuția.

Dacă doriți să părăsiți editorul fără salvarea modificărilor, folosiți comanda **q**.

Rezumatul comenzilor ed

Sesiunea demonstrativă v-a prezentat modul de utilizare a comenzilor **ed** pentru a efectua operațiile de bază necesare la editarea unui fișier text. Tabelul 24-1 prezintă toate comenzile **ed** folosite frecvent:

Tabelul 24-1 Comenzi ed folosite uzual

Comandă	Semnificație
!command	Execută o comandă de interpretor
\$	Se deplasează la ultima linie din buffer
%	Aplică tuturor liniilor din buffer comanda ce urmează (de exemplu, %p afișează toate liniile)
+	Se deplasează la linia următoare
+N	Se deplasează la a <i>N</i> -a linie următoare (<i>N</i> este un număr)
,	Aplică tuturor liniilor din buffer comanda care urmează (de exemplu, p afișează toate liniile); asemănătoare cu %
-	Se deplasează la linia precedentă

(continuare)

Comandă	Semnificație
<i>-N</i>	Se deplasează la a <i>N</i> -a linie anterioară (<i>N</i> este un număr)
.	Se referă la linia curentă din buffer
<i>/regex/</i>	Caută în liniile următoare expresia regulată specificată; vezi capitolul 6 pentru detalii despre expresii regulate
;	Se referă la un grup de linii, de la cea curentă până la ultima din buffer
=	Afișează numărul liniei
<i>?regex?</i>	Caută în liniile precedente expresia regulată specificată; vezi capitolul 6 pentru detalii despre expresii regulate
^	Se deplasează la linia precedentă
<i>^N</i>	Se deplasează la a <i>N</i> -a linie anterioară (unde <i>N</i> este un număr); de asemenea vezi comanda <i>-N</i>
a	Adaugă la sfârșitul liniei curente
c	Schimbă liniile specificate
d	Șterge liniile specificate
e <i>file</i>	Editează fișierul
f <i>file</i>	Schimbă numele fișierului prestabilit
h	Afișează explicația ultimei erori
H	Activează raportarea erorilor în modul comunicativ
i	Inserează text înaintea liniei curente
j	Unește liniile contigue
kx	Marchează linia cu litera x (această linie poate fi referită ulterior 'x')
l	Tipărește (afișează) liniile
m	Deplasează linii
N	Se deplasează la linia N
Newline	Afișează linia următoare și o transformă în linie curentă
P	Comută modul prompt pe on sau off
q	Părăsește editorul
Q	Părăsește editorul fără salvarea modificărilor
r <i>file</i>	Citește și inserează conținutul fișierului după linia curentă
s/old/new/	Înlocuiește șirul <i>old</i> cu <i>new</i>
Space <i>N</i>	Un spațiu, urmat de <i>N</i> ; a <i>N</i> -a linie înainte (<i>N</i> este un număr)
u	Anulează ultima comandă
w <i>file</i>	Adaugă conținutul bufferului la sfârșitul fișierului specificat
w <i>file</i>	Salvează bufferul în fișierul specificat (dacă nu este denumit nici un fișier, salvează în fișierul prestabilit – fișierul al cărui conținut tocmai este editat de vi)



Sugestie

2a

(Tastati liniile de text. Marcați sfârșitul a ceea ce aveți de introdus tastând un singur punct pe o linie separata.)

Pentru a afișa liniile de la 3 la 15, folosiți această comandă:

```
3,15p
```

Cu toate că este posibil să nu folosiți deseori editorul *ed*, multe dintre comenzile acestuia sunt valabile și în cazul editorului *vi*. Așa cum se arată în următoarea secțiune, editorul *vi* acceptă (în modul de comandă) comenzile pe care le folosește *ed*.

Folosirea editorului vi

vi este un editor de text orientat spre pagină care vă permite să vizualizați simultan mai multe linii dintr-un fișier. Majoritatea sistemelor UNIX, inclusiv Linux, sunt livrate cu acest editor. Prin urmare, dacă învățați caracteristicile de bază ale editorului *vi*, veți fi capabil să editați fișiere text pe aproape orice sistem UNIX.

Ca și *ed*, editorul *vi* lucrează cu un buffer. Când *vi* editează un fișier, el citește fișierul într-un buffer – un bloc de memorie – și vă permite să modificați textul din buffer. Editorul *vi* folosește, de asemenea, fișiere temporare în timpul editării, dar fișierul original nu este modificat până când aceste modificări nu sunt salvate cu o comandă specifică *vi*.

Setarea tipului terminalului

Înainte de a porni un editor pe tot ecranul precum *vi*, trebuie să setați variabila de mediu *TERM* în funcție de tipul terminalului (precum *vt100* sau *xterm*). Editorul *vi* folosește tipul terminalului pentru a căuta caracteristicile terminalului în fișierul */etc/termcap* și apoi controlează terminalul în modul full-screen (pe tot ecranul).

Când rulați *X*, puteți folosi *vi* într-o fereastră *xterm*. Tipul terminalului pentru o fereastră *xterm* este *xterm*. Când porniți *xterm*, el setează automat variabila de mediu *TERM* pe *xterm*. Prin urmare, veți putea folosi *vi* într-o fereastră *xterm* fără să setați explicit variabila *TERM*.

Lansarea editorului vi

Pe sistemul dumneavoastră Linux, editorul *vim* emulează ceea ce apare ca editor *vi* pe majoritatea sistemelor UNIX. Puteți, totuși, să continuați folosirea numelui *vi* pentru a invoca editorul, deoarece există o legătură simbolică între *vi* și *vim*. Pentru a vedea această legătură, tastează următoarea comandă *ls*:

```
ls -l /usr/bin/vi
lrwxrwxrwx 1 root root 13 Mar 11 15:04 /usr/bin/vi ->
.. / .. /Bin/vim
```

Dacă doriți să consultați paginile manualului online pentru *vi*, tastează următoarea comandă:

```
man vi
```

Pentru a lansa editorul, folosiți numele *vi* și apelați-l cu următoarea comandă:

```
vi [flags] [+cmd] [filename]
```

Argumentele aflate între parantezele pătrate sunt opționale. Următoarea listă explică argumentele:

- **flags** sunt indicatori alcătuiți dintr-un singur caracter, care controlează modul în care rulează **vi**;
- **+cmd** determină **vi** să execute comanda specificată după lansare. (Veți învăța mai multe despre comenzi în secțiunea „Rezumatul comenzilor **vi**“.)

- **filename** este numele fișierului ce va fi editat.

Argumentele **flags** pot fi unul sau mai multe dintre următoarele:

- **-c cmd** execută comanda specificată înainte de începerea editării.
- **-e** pornește în modul de comandă două puncte (:) (descriș în secțiunea următoare).
- **-i** pornește în modul introducere text (descriș în secțiunea următoare).
- **-m** determină editorul să caute prin fișier ceva asemănător cu un mesaj de eroare emis de un compilator
- **-R** protejează fișierele la scriere, pentru a nu fi suprascrise accidental.
- **-s** rulează în modul de securitate maximă, în care multe comenzi potențial periculoase sunt dezactivate.
- **-v** pornește în modul de comandă vizuală (descriș în secțiunea următoare).

Totuși, de cele mai multe ori se lansează **vi** cu un nume de fișier ca unic argument, astfel:
vi /etc/hosts

Altă modalitate obișnuită de a porni editorul **vi** este de a trece la un anumit număr de linie chiar de la început. Pentru a începe editarea de la linia 296 a fișierului **/etc/X11/XF86Config**, de exemplu, folosiți următoarea comandă:

vi +296 /etc/X11/XF86Config

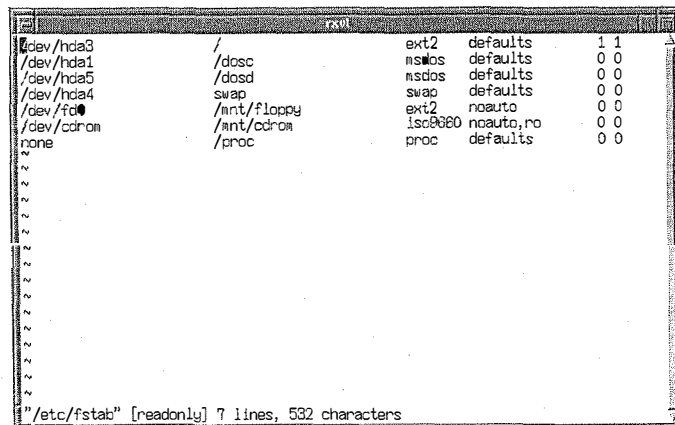
Această modalitate de a lansa **vi** este folositoare atunci când editați un fișier sursă după ce compilatorul raportează o eroare la un anumit număr de linie.

Conceptele editorului vi

Când editați un fișier, editorul **vi** îl încarcă într-un buffer, îi afișează primele linii într-o fereastră pe tot ecranul și poziționează cursorul pe prima linie. Când tastați comanda **vi /etc/fstab** într-o fereastră **xterm**, de exemplu, obțineți o fereastră text pe tot ecranul, așa cum se vede în figura 24-1.

Figura 24-1

Un fișier afișat într-o fereastră text pe tot ecranul de către editorul **vi**.



Ultima linie prezintă informații despre fișier, inclusiv numărul de linii și de caractere. Apoi, această zonă este folosită pentru introducerea comenzilor. Restul liniilor sunt folosite pentru a afișa fișierul. Dacă acesta conține mai puține linii decât fereastra, **vi** afișează liniile vide cu o tildă (~) în prima coloană.

Linia curentă este marcată de cursor, care apare ca un mic dreptunghi negru. Cursorul apare în partea superioară a unui caracter. În figura 24-1, cursorul se află pe primul caracter al primei linii.

În **vi** puteți lucra într-unul din următoarele trei moduri:

- **Modul comandă vizuală** este prestabilit. În acest mod, orice tastați este interpretat ca o comandă ce se aplică liniei pe care se află cursorul. Comenzile **vi** sunt asemănătoare cu cele din **ed** și sunt prezentate în secțiunea „Rezumatul comenzilor **vi**“.

- **Modul comandă două puncte (:)** vă permite să citiți sau să scrieți fișiere, să setați opțiuni **vi** și să părăsiți editorul. Toate comenzile din acest mod încep cu caracterul două puncte (:). Când introduceți acest caracter, **vi** poziționează cursorul pe ultima linie și vă permite să tastați o comandă. Aceasta intră în execuție când apăsați **Enter**.

- **Modul introducere text** vă permite să introduceți text într-un fișier. Pentru a comuta pe modul introducere text folosiți comanda **a** (inserează după cursor), **A** (adaugă la sfârșitul liniei) sau **i** (inserează după cursor). După introducerea liniilor de text, trebuie să tastați **Esc** pentru a comuta din modul introducere text în modul comandă vizuală.

O problemă cu aceste trei moduri o constituie faptul că nu puteți determina cu ușurință modul curent al editorului. Este frustrant să începeți să tastați și să vă dați seama că **vi** nu este în modul introducere text. Situația inversă este de asemenea obișnuită: puteți fi în mod text și să introduceți de fapt o comandă. Pentru a fi siguri că **vi** este în mod comandă, apăsați **Esc** de câteva ori. (Această acțiune nu are nici un efect negativ.)



Pentru a vizualiza asistența online în **vi**, tastați **:help** în timp ce sunteți în modul comandă.

Sugestie

Examinarea unei sesiuni demonstrative cu editorul vi

Pentru a începe editarea fișierului **/etc/fstab**, introduceți următoarea comandă:

vi /etc/fstab

Figura 24-1 prezintă ecranul rezultat, cu primele linii ale fișierului afișate într-o fereastră text pe tot ecranul. Ultima linie arată numele fișierului și o statistică: numărul de linii și de caractere.

Editorul **vi** poziționează inițial cursorul pe primul caracter. Printre primele lucruri pe care trebuie să le învățați este modul de deplasare a cursorului. Mai întâi, încercați următoarele comenzi (fiecare comandă este formată dintr-o singură literă; tastați litera iar **vi** va reacționa corespunzător):

- **j** deplasează cursorul cu o linie în jos
- **k** deplasează cursorul cu o linie în sus
- **h** deplasează cursorul cu un caracter la stânga
- **l** deplasează cursorul cu un caracter la dreapta

În loc de a vă deplasa cu o linie sau cu un caracter o dată, vă puteți deplasa cuvânt cu cuvânt. Încercați următoarele comenzi alcătuite dintr-un singur caracter pentru deplasări ale cursorului la nivel de cuvânt:

- w deplasează cursorul cu un cuvânt înainte
- b deplasează cursorul cu un cuvânt înapoi

Ultimul tip de deplasare a cursorului este de tip linii cu linii. Încercați următoarele comenzi pentru a vedea ce se întâmplă:

- Ctrl-D deplasează cursorul în jos cu jumătate de ecran
- Ctrl-U deplasează cursorul în sus cu jumătate de ecran

Ultimele două comenzi, desigur, nu sunt necesare când fișierul conține doar câteva linii. Când editați fișiere mari, acest tip de deplasare este util.

În orice moment, vă puteți deplasa la un anumit număr de linie. În această situație intervine caracterul două puncte (:). Pentru a vă deplasa la linia 1, de exemplu, tasteați următoarele și apoi apăsați Enter:

```
:1
```

Când tasteați caracterul două puncte (:), vi îl afișează pe ultima linie a ecranului. Din acel moment, vi folosește textul pe care-l tasteați drept comandă. Trebuie să apăsați Enter pentru a transmite editorului vi comanda. În modul de comandă două puncte (:), vi acceptă toate comenzile editorului ed, precum și alte comenzi.

Pentru a căuta un șir, mai întâi tasteați un slash (/). Editorul vi afișează caracterul slash pe ultima linie a ecranului. Tasteați șirul de căutat și apoi apăsați Enter. Editorul vi localizează șirul și poziționează cursorul la începutul acestuia. Astfel, pentru a localiza șirul `dosc` în fișierul `/etc/fstab`, tasteați următoarele:

```
/Dosc
```

Pentru a șterge linia pe care este cursorul, tasteați **dd** (două litere d mic). Editorul vi șterge acea linie de text și o transformă pe următoarea în linie curentă.



Sugestie

Pentru a începe introducerea textului în fața cursorului, tasteați **i** (un i mic). Editorul vi comută pe modul introducere text. Acum puteți introduce text. Când terminați, apăsați Esc pentru a reveni la modul comandă vizuală.

După ce terminați editarea fișierului, puteți salva modificările folosind comanda `:w`. Dacă doriți să salvați modificările și să ieșiți, puteți tastea `:wq` pentru a executa ambii pași în același timp. Editorul vi salvează modificările în fișier și își încheie execuția.

Dacă doriți să părăsiți editorul fără salvarea modificărilor, folosiți comanda `:q!`.

Rezumatul comenzilor vi

Sesiunea de editare demonstrativă este utilă în înțelegerea comenzilor vi, în special despre cele trei moduri:

- modul comandă vizuală (prestabilit)
- modul introducere text, pe care-l activați tastând **a**, **A** sau **i**
- modul comandă cu două puncte (:), în care introduceți comenzile urmate de caracterul două puncte (:)

Pe lângă cele câteva comenzi prezentate în sesiunea demonstrativă, vi acceptă multe alte comenzi. Tabelul 24-2 prezintă comenzile vi de bază, organizate după sarcinile pe care le îndeplinesc.

Tabelul 24-2 Comenzi vi de bază

Comandă	Semnificație
Inserare text	
a	Inserează text după cursor
A	Inserează text la sfârșitul liniei curente
I	Inserează text la începutul liniei curente
i	Inserează text înaintea cursorului
o	Inserează o linie vidă sub linia curentă
O	Inserează o linie vidă deasupra liniei curente
Ctrl-v	Inserează orice caracter special în modul introducere text
Ștergere text	
D	Șterge toate caracterele de la cursor până la sfârșitul liniei curente
dd	Șterge linia curentă
dw	Șterge toate caracterele de la cursor până la sfârșitul următorului cuvânt
x	Șterge caracterul pe care se află cursorul
Modificare text	
C	Modifică textul cuprins între cursor și sfârșitul liniei curente
cc	Modifică linia curentă
cw	Modifică cuvântul
J	Unește linia curentă cu următoarea
rx	Înlocuiește caracterul pe care se află cursorul cu x (x este orice caracter)
~	Schimbă caracterul pe care se află cursorul din litera mică în majusculă sau invers
Deplasări cursor	
\$	Deplasează cursorul la sfârșitul liniei curente
;	Repetă ultima comandă f sau F
^	Deplasează cursorul la începutul liniei curente
e	Deplasează cursorul la sfârșitul cuvântului curent
fx	Deplasează cursorul la prima apariție a caracterului x pe linia curentă
Fx	Deplasează cursorul la ultima apariție a caracterului x pe linia curentă
H	Deplasează cursorul la începutul ecranului
h	Deplasează cursorul cu un caracter la stânga
j	Deplasează cursorul cu o linie în jos
k	Deplasează cursorul cu o linie în sus
L	Deplasează cursorul la sfârșitul ecranului

(continuare)

Tabelul 24-2 Continuare

Comandă	Semnificație
l	Deplasează cursorul cu un caracter la dreapta
M	Deplasează cursorul în mijlocul ecranului
N	Deplasează cursorul la coloana n din linia curentă
NG	Plasează cursorul pe linia n
w	Deplasează cursorul la începutul următorului cuvânt
Marcarea unei locații	
' x	Deplasează cursorul la începutul liniei ce conține semnul x
`x	Deplasează cursorul la semnul x
mx	Marchează locația curentă cu litera x
Derulare text	
Ctrl-b	Derulează textul înapoi cu un ecran
Ctrl-d	Derulează textul înainte cu jumătate de ecran
Ctrl-f	Derulează textul înainte cu un ecran
Ctrl-u	Derulează textul înapoi cu jumătate de ecran
Reîmprospătare ecran	
Ctrl-l	Redesenează ecranul
Decupare și lipire text	
" xNdd	Șterge n linii și le plasează în bufferul x (x reprezintă orice caracter scris cu literă mică)
"XNyy	La fel ca "xNyy, cu excepția faptului că liniile extrase sunt adăugate la sfârșitul bufferului x
"xNyy	Extrage n linii (N este un număr) dintr-un buffer x (x reprezintă orice caracter scris cu majusculă)
"xp	Inserează liniile extrase din bufferul x după linia curentă
P	Inserează liniile extrase deasupra liniei curente
p	Inserează liniile extrase sub linia curentă
yy	Copiază linia curentă într-un buffer fără nume.
Comenzi cu două puncte (:)	
!:command	Execută o comandă de interpretor
:e filename	Editează fișierul
:f	Afișează numele fișierului și numărul liniei curente
:N	Se deplasează la linia n (n este un număr)
:q	Părăsește editorul
:q!	Părăsește editorul fără salvarea modificărilor

Comandă	Semnificație
:r filename	Citește fișierul și-l inserează după linia curentă
:w filename	Scrie un buffer într-un fișier
:wq	Salvează modificările și părăsește editorul
Căutare text	
/string	Caută înainte șirul string
?string	Caută înapoi șirul string
n	Găsește următorul șir
Vizualizare informații despre fișier	
Ctrl-g	Aată numele și mărimea fișierului și numărul liniei curente
Diverse	
u	Anulează ultima comandă
Esc	Comută de pe modul introducere pe modul comandă vizuală
U	Anulează schimbările recente ale liniei curente

Lucrul cu editorul GNU Emacs

Editoarele de texte sunt o chestiune de preferință personală și mulți utilizatori UNIX preferă GNU Emacs. Cu toate că la început intimidează pe mulți, GNU Emacs este unul din acele programe care vă cucerește pe măsură ce-l folosiți; are atât de multe facilități, încât mulți utilizatori și programatori își realizează toate operațiile direct din acest editor.

Un avantaj semnificativ al editorului GNU Emacs îl reprezintă adaptabilitatea lui pe aproape orice sistem de operare imaginabil, de la MS-DOS la UNIX. Dacă pe un anumit sistem nu dispuneți de GNU Emacs, îl puteți procura dintr-o multitudine de surse din Internet. Pentru sistemul dumneavoastră Linux, puteți prelua editorul GNU Emacs de pe CD-ROM-ul anexat acestei cărți; puteți opta pentru instalarea simultană a GNU Emacs și Linux pe sistemul dumneavoastră. Dacă tocmai ați început să folosiți editoare de text UNIX, v-aș recomanda să învățați GNU Emacs. Astfel, dobândiți cunoștințe pe care le puteți folosi pe orice sistem UNIX.



Observație

Deoarece GNU Emacs este atât de puternic și versatil, descrierea detaliată programului ocupă minim o carte. De fapt, există pe piață foarte multe cărți despre GNU Emacs. O'Reilly & Associates a publicat o carte utilă despre GNU Emacs numită *Learning GNU Emacs* – Învățarea GNU Emacs.

Pe un terminal text, GNU Emacs rulează în modul text pe tot ecranul. În sistemul X, GNU Emacs rulează într-o fereastră. În oricare dintre moduri, comenzile de bază rămân aceleași. Versiunea X vă permite, de asemenea, să poziționați cursorul cu mouse-ul.

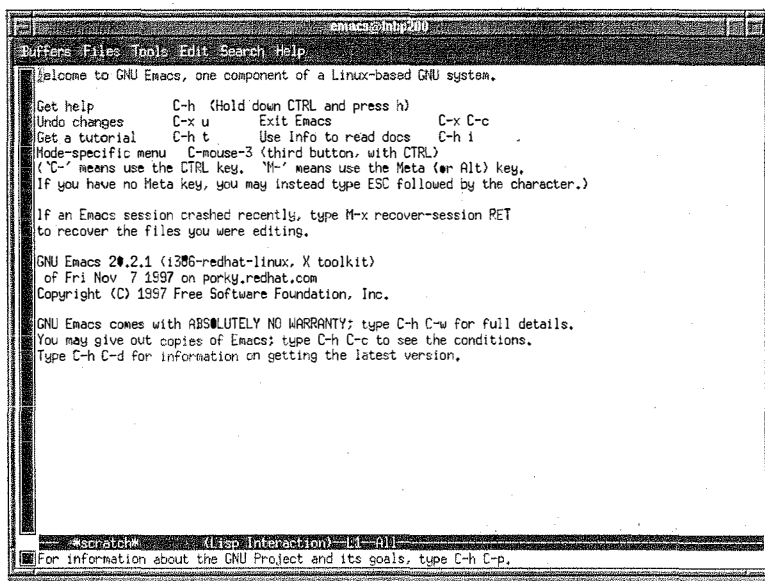
Următoarele secțiuni prezintă o scurtă introducere a facilităților de editare a textului în GNU Emacs.

Lansarea editorului GNU Emacs

Pe un sistem Linux puteți porni editorul GNU Emacs tastând **emacs**. Dacă rulați X, tastați **emacs&** într-o fereastră **xterm** pentru a lansa GNU Emacs. Această comandă lansează versiunea X Window System a programului GNU Emacs în fundal și vă permite să vă continuați activitatea într-o fereastră **xterm**.

Când este lansat pentru prima dată, editorul creează un buffer denumit **scratch** și afișează într-o fereastră un mesaj de asistență, urmat de unul de copyright, ca în figura 24-2.

Figura 24-2
Fereastra inițială afișată de versiunea X a editorului GNU Emacs.



Fereastra inițială a editorului GNU Emacs vă prezintă de asemenea informații folositoare în suprafața de editare a conținutului fișierului, informații care vă oferă următoarele sugestii:

- Apăsați **C-h t** (apăsați **Ctrl-h** și apoi **t**) pentru a vedea un îndrumar online despre GNU Emacs.
- Apăsați **C-h i** pentru a lansa **info**, sistemul de asistență online pentru toate produsele software GNU.

Capitolul 22 descrie modul de utilizare al sistemului **info** (Sugestie: Tastați **m** și apoi **emacs** pentru a vedea informații despre GNU Emacs.)

De asemenea, mesajul vă indică faptul că, dacă vreți să părăsiți editorul GNU Emacs, trebuie să tastați **C-x C-c** (adică **Ctrl-X** urmat de **Ctrl-C**). Așa cum veți învăța în secțiunea „Tastarea comenzilor GNU Emacs”, toate combinațiile de taste pentru comenzile GNU Emacs încep cu un caracter de control sau **Escape** (care mai este denumit **Meta** și prescurtat **M** în documentația GNU Emacs).



Observație

Pentru a fi consecvent cu notațiile din GNU Emacs, voi folosi **C-x** pentru **Ctrl-x** și **M-x** pentru **Escape-x** (**x** este orice caracter).

Elementele de bază ale editorului GNU Emacs

Ani de zile am folosit foarte mult GNU Emacs, dar cred că n-am văzut decât vârful aisbergului, având în vedere toate facilitățile acestui editor. De aceea, cred că cel mai bine este să începeți cu un mic subset de comenzi GNU Emacs. Apoi, o dată cu trecerea timpului, vă puteți îmbunătăți treptat repertoriul de comenzi și caracteristici ale acestui editor.

GNU Emacs este construit în jurul câtorva concepte de bază, utile de învățat. Iată câteva dintre acestea:

- Ca și alte editoare de text (**vi** sau **ed**), GNU Emacs folosește un buffer pentru a stoca textul introdus și apoi modificat. Pentru a actualiza conținutul unui fișier trebuie să salvați în mod explicit bufferul.
- Spre deosebire de editoarele **vi** și **ed**, GNU Emacs nu necesită tastarea unei comenzi speciale pentru a introduce text într-un buffer. În mod prestabilit, toate datele tastate vor fi introduse într-un buffer.
- GNU Emacs folosește un cursor de forma unui bloc. Când tastați text, editorul îl inserează în fața caracterului pe care se află cursorul.
- GNU Emacs are nume de comenzi lungi, descriptive, care sunt legate de combinații de taste specifice – acestea sunt *asocierile de taste* (key bindings) pentru comenzile GNU Emacs. **C-x C-c**, de exemplu, este legată de comanda GNU Emacs **save-buffers-kill-emacs**.
- Toate asocierile de taste din editorul GNU Emacs încep cu un caracter de control (pentru care apăsați simultan **Ctrl** și un caracter) sau cu **Escape**.
- GNU Emacs folosește câteva moduri, fiecare din ele furnizând un anumit tip de mediu de editare. (În modul **C** de exemplu, editorul vă ajută prin decalarea parantezelor acolade.)
- În fereastra editorului GNU Emacs, ultima linie de pe ecran este denumită *minibuffer*; ea afișează toate comenzile și numele de fișiere pe care le tastați. Linia a doua de jos este denumită *mode line* – linie de mod. Pe această linie, GNU Emacs afișează numele bufferului și modul curent (modul prestabilit este denumit *fundamental*).
- Nu trebuie să porniți GNU Emacs de fiecare dată când doriți să editați un fișier. Porniți editorul și apoi deschideți unul sau mai multe fișiere pentru editare. Salvați și închideți fișierele pe care le-ați editat.
- Puteți folosi mai multe buffere în GNU Emacs și puteți executa operația de decupare și lipire între buffere.

Tastarea comenzilor editorului GNU Emacs

GNU Emacs dispune de un set cuprinzător de comenzi, fiecare având un nume lung, descriptiv, astfel:

save-buffer save-buffers-kill-emacs scroll-up previous-line

Totuși, majoritatea comenzilor sunt asociate cu combinații de taste oarecum criptice. Altfel, în loc să editați ați pierde timpul cu tastarea acestor comenzi lungi.



Observație

Cu toate că puteți introduce oricare din comenzile descriptive în minibuffer (în partea de jos a ferestrei editorului GNU Emacs), mijloacele de bază de introducere a comenzilor rezidă în combinațiile de taste speciale. Aceste combinații de taste încep cu unul din următoarele caractere:

- Un caracter de control pe care-l introduceți apăsând simultan tasta **Ctrl** și caracterul respectiv. În documentația editorului GNU Emacs, fiecare caracter de control este prescurtat sub forma **C-x** (**x** este o literă). Această carte folosește notația **Ctrl-x** pentru a indica caracterul de control ce corespunde literei **x**. În manualul online al editorului GNU Emacs, **Ctrl-v** este scris **C-v**.
- Un caracter **Escape**. În GNU Emacs, tasta **Escape** este prescurtată drept tastă **Meta**, sau **M**. Astfel, combinația **Esc v** (**Escape** urmată de litera **v**) este scrisă **M-v**.

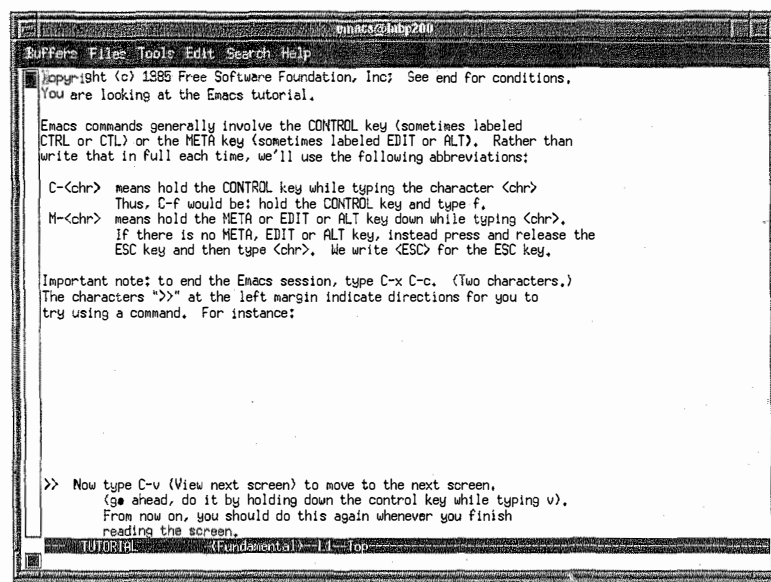
De cele mai multe ori veți introduce comenzile de control, care necesită apăsarea tastei Ctrl împreună cu o literă. Ctrl-v sau C-v, de exemplu, determină GNU Emacs să se deplaseze cu un ecran înainte.

Comenzile cu prefixul Esc sunt mai ușor de introdus, deoarece trebuie să apăsați tastele succesiv: mai întâi tastați Escape și apoi litera. Esc v sau M-v, de exemplu, determină GNU Emacs să se deplaseze înapoi cu un ecran de text. Pentru a introduce această comandă, apăsați mai întâi Escape și apoi v. Deși comenzile Ctrl și Esc par complicate, puteți învăța foarte repede un set de comenzi de bază.

Obținerea de asistență

Cea mai bună sursă de informație despre GNU Emacs este chiar editorul însuși. Pentru începători, GNU Emacs conține un îndrumar online care vă prezintă chestiuni de bază despre acesta. Pentru a folosi îndrumarul, tastați C-h t. Editorul afișează ecranul inițial al îndrumarului în fereastra lui, așa cum se arată în figura 24-3.

Figura 24-3
GNU Emacs afișează îndrumarul online când tastați C-h t.



Așa cum arată instrucțiunile din partea de jos a figurii 24-3, îndrumarul vă prezintă etapele și vă recomandă să folosiți comenzile editorului. Dacă sunteți începător în folosirea acestui editor, ar trebui să parcurgeți îndrumarul. Deoarece îndrumarul solicită implicarea efectivă a utilizatorului, vă veți face o idee despre GNU Emacs după ce îl parcurgeți.

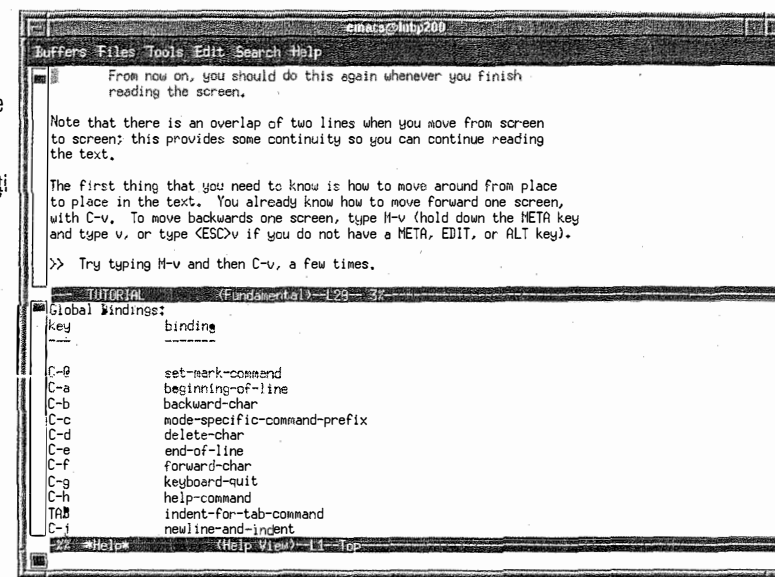
Pe lângă îndrumar, puteți căuta asocierile de taste pentru diferite comenzi din GNU Emacs. Pentru a vedea asocierile de taste, tastați C-h b. Editorul împarte fereastra în două și afișează o listă cu asocierile de taste în partea de jos, așa cum se vede în figura 24-4.

Fiecare linie din lista cu asocieri de taste arată numele tastei și comanda GNU Emacs asociată cu aceasta. Dacă apăsați tasta respectivă, editorul execută comanda asociată acesteia.

Figura 24-4 ilustrează o altă caracteristică a editorului GNU Emacs: posibilitatea de a-și împărți fereastra în două sau mai multe părți. După împărțire, fiecare parte devine o fereastră separată, cu propriul ei buffer.

Figura 24-4

Editorul GNU Emacs afișează legăturile de taste într-o fereastră împărțită în două atunci când tastați C-h b.



În figura 24-4, puteți derula lista asocierilor de taste cu următoarele comenzi:

- C-x o comută în fereastra cealaltă (fereastra de jos, care conține bufferul de asistență cu lista acestuia de asocieri de taste)
- C-v derulează conținutul bufferului de asistență

Tabelul 24-3 listează alte comenzi de asistență ale editorului GNU Emacs.

Tabelul 24-3 Comenzi de asistență ale editorului GNU Emacs

Comandă	Semnificație
C-h c	Vă solicită o succesiune de taste, pe care apoi o descrie pe scurt
C-h f	Vă solicită o comandă GNU Emacs, pe care o descrie
C-h k	Vă solicită o succesiune de taste și descrie operația executată de aceasta
C-h l	Afișează ultimele 100 de caractere tastate
C-h m	Afișează modul curent al editorului GNU Emacs
C-h s	Afișează tabela de sintaxă pentru bufferul curent
C-h v	Vă solicită numele unei variabile, pe care apoi o descrie
C-h w	Vă solicită o comandă și afișează asocierile de taste pentru aceasta

Citirea unui fișier

După ce porniți editorul GNU Emacs, puteți citi un fișier folosind comanda `C-x C-f`. Pentru a deschide fișierul `/usr/src/linux/COPYING`, de exemplu, parcurgeți următorii pași:

1. Tastați **C-x C-f**. GNU Emacs vă solicită numele unui fișier.
2. Tastați numele fișierului – `/usr/src/linux/COPYING` – și apăsați Enter. GNU Emacs citește fișierul într-un buffer pe care-l afișează într-o fereastră.

Pentru a deschide un fișier din directorul curent, nu trebuie să tastați numele întreg al directorului; tastați doar numele fișierului. În mod prestabilit, editorul GNU Emacs caută fișierul în directorul de lucru curent.



Sugestie În GNU Emacs nu trebuie să tastați numele întreg al fișierului; introduceți doar primele caractere și apoi apăsați Tab. Dacă numele parțial identifică în mod unic fișierul, GNU Emacs completează numele. Astfel, trebuie să tastați doar primele caractere ale numelui fișierului – suficiente pentru a identifica în mod unic fișierul. Puteți folosi această facilități pentru a evita introducerea numelor lungi de fișiere.

Să presupunem că directorul curent conține fișierul `Makefile`, care este singurul al cărui nume începe cu subșirul `Mak`. Pentru a încărca acest fișier în GNU Emacs, apăsați `C-x C-f`, tastați `Mak` și apoi apăsați Tab. GNU Emacs completează numele fișierului și citește fișierul `Makefile`.

Deplasarea într-un buffer

Unul dintre primele lucruri pe care trebuie să-l faceți este să vă deplasați într-un buffer. Cursorul marchează locul curent din buffer; datele tastate sunt introduse în buffer înaintea caracterului pe care se află cursorul. Astfel, pentru a insera text într-un fișier, trebuie să citiți fișierul, să deplasați cursorul în locul dorit și să introduceți textul.

Pentru a vă deplasa în buffer, trebuie să deplasați cursorul. Pentru a deplasa cursorul în orice direcție trebuie să folosiți tastele de control. Iată care sunt cele șase comenzi de bază pentru deplasarea cursorului:

- `C-b` deplasează cursorul înapoi cu un caracter
- `C-f` deplasează cursorul înainte cu un caracter
- `C-n` deplasează cursorul pe linia următoare (în timp ce încearcă să păstreze aceeași poziție a coloanei ca în linia curentă)
- `C-p` deplasează cursorul pe linia precedentă (în timp ce încearcă să păstreze aceeași poziție a coloanei ca în linia curentă)
- `C-a` deplasează cursorul la începutul liniei curente
- `C-e` deplasează cursorul la sfârșitul liniei curente

Dacă deplasarea caracter cu caracter este prea lentă, vă puteți deplasa cuvânt cu cuvânt, folosind următoarele comenzi:

- `M-f` deplasează cursorul cu un cuvânt înainte
- `M-b` deplasează cursorul cu un cuvânt înapoi

Vă puteți deplasa chiar cu intervale mai mari într-un buffer. Următoarele comenzi vă permit să vă deplasați ecran cu ecran:

- `C-v` deplasează cursorul înainte cu un ecran
- `M-v` deplasează cursorul înapoi cu un ecran

Puteți folosi alte două comenzi simple pentru salturi într-adevăr mari ale cursorului:

- `M-<` (Escape, urmat de tasta „mai mic“) deplasează cursorul la începutul bufferului
- `M->` (Escape, urmat de tasta „mai mare“) deplasează cursorul la sfârșitul bufferului

O facilități a editorului GNU Emacs vă permite să economisiți timp prin repetarea unei comenzi de un număr specificat de ori. Să presupunem că doriți să vă deplasați cu 13 caractere înainte. Puteți face acest lucru tastând `C-f` de 13 ori sau tastând **C-u 13 C-f**. Comanda `C-u` acceptă un contor care repetă comanda ce urmează de numărul respectiv de ori. Dacă nu introduceți un contor și tastați pur și simplu o comandă după `C-u`, GNU Emacs repetă comanda de patru ori. Astfel, `C-u C-f` înseamnă „deplasare înainte cu patru caractere“.



Secret

Există o singură excepție în modul de comportare al comenzii `C-u`: Când folosiți `C-u` cu comanda `M-v` sau `C-v`, GNU Emacs nu repetă comenzile de derulare a ecranului. În schimb, GNU Emacs derulează ecranul în sus sau în jos, în funcție de contorul specificat. Astfel, dacă tastați `C-u 10 C-v`, editorul deplasează cursorul 10 linii în jos (nu 10 ecrane).

Inserarea și ștergerea textului

Deoarece GNU Emacs nu are moduri speciale de inserție și comandă ca editorul `vi`, pentru a insera text în GNU Emacs pur și simplu începeți să tastați. Editorul înserează textul în fața cursorului. GNU Emacs, desigur, interpretează caracterele de control și Esc drept începutul unei comenzi.

Puteți profita de avantajul posibilității de a repeta de un număr de ori o comandă pentru a insera multiple copii ale unui caracter. Eu folosesc de obicei următoarea linie pentru a separa secțiunile unui program C:

```
/*-----*/
```

Introduc această linie apăsând următoarele taste:

```
/* C-u 51 - */
```

Ignorați spațiile; eu nu le titez. Titez mai întâi `/*`. Apoi titez **C-u 51**. În final, titez liniuța `(-)`, urmată de `*/`.

Puteți șterge text în GNU Emacs în următoarele moduri:

- Pentru a șterge caracterul din fața cursorului, apăsați tasta Delete.
- Pentru a șterge caracterul pe care se află cursorul, tastați `C-d`.
- Pentru a șterge cuvântul situat după cursor, tastați `M-d`.
- Pentru a șterge cuvântul situat imediat înaintea cursorului, tastați `M-Delete`.
- Pentru a șterge textul de la cursor până la sfârșitul liniei, tastați `C-k`. (GNU Emacs denumește această comandă „kill the line“ – distrugerea liniei.)



Sugestie

Când ștergeți mai mult decât un caracter, GNU Emacs salvează ceea ce ați șters. Pentru a obține textul șters, pur și simplu tastați `C-y`.

Puteți anula o modificare folosind comanda `C-x u`. De fiecare dată când tastați `C-x u`, editorul execută operația de anulare pentru o comandă anterioară. Pentru a anula efectul ultimelor două comenzi, de exemplu, tastați `C-x` de două ori.

Căutarea și înlocuirea

Fiecare editor de text are facilități de căutare și înlocuire, iar GNU Emacs nu face excepție. Cele două comenzi de căutare cele mai utilizate sunt

- C-s string: caută incremental înainte șirul string
- C-r string: caută incremental înapoi șirul string

Când tastați C-s pentru a căuta înainte, GNU Emacs vă solicită să introduceți un șir de căutat în minibuffer (ultima linie din fereastra GNU Emacs). Pe măsură ce introduceți caracterele șirului, GNU Emacs „sare” la prima apariție a șirului pe care l-ați tastat până în acel moment. Până terminați de tastat șirul de căutat, GNU Emacs poziționează cursorul la sfârșitul următoarei apariții a șirului de căutat (cu condiția ca, desigur, să găsească șirul).

Pentru a găsi următoarea apariție a șirului, pur și simplu tastați C-s din nou. Pentru a termina căutarea, apăsați Enter. Puteți, de asemenea, să opriți căutarea folosind comenzile pentru controlul cursorului, precum C-f sau C-b.

Căutarea în direcție inversă decurge asemănător; tastați C-r în loc de C-s.

Editorul GNU Emacs vă permite, de asemenea, să înlocuiți o apariție a unui șir cu altul. Cele două comenzi de bază pentru înlocuirea șirurilor sunt replace-string și query-replace. Comanda replace-string înlocuiește toate aparițiile unui șir cu altul. Comanda query-replace lucrează asemănător, dar editorul GNU Emacs vă avertizează de fiecare dată când este pe cale să înlocuiască un șir, permițându-vă să decideți asupra înlocuirilor.

Comanda query-replace este legată de succesiunea de taste M-% (Escape urmat de %). Pentru a executa o operație query-replace, mai întâi apăsați M-%. Editorul afișează următorul prompt în minibuffer:

Query replace:

Introduceți șirul pe care vreți să-l înlocuiți – să presupunem că este 1998 – și apoi apăsați Enter. GNU Emacs vă solicită să introduceți șirul cu care vreți să-l înlocuiți pe cel indicat anterior, astfel: Query replace 1998 with:

Introduceți șirul cu care vreți să-l înlocuiți – să spunem 1999 – și apăsați Enter. GNU Emacs deplasează cursorul la următoarea apariție a șirului ce trebuie înlocuit și afișează un prompt în minibuffer, astfel:

Query replacing 1998 with 1999: (? for help)

Tastați y sau apăsați tasta spațiu pentru a permite editorului să înlocuiască șirul. Altfel, tastați n sau Delete pentru a opri GNU Emacs să înlocuiască șirul. În oricare din cazuri, GNU Emacs se deplasează la următoarea apariție a șirului și repetă promptul. Când nu au mai rămas șiruri, editorul afișează un mesaj în minibuffer, informându-vă câte apariții ale șirului au fost înlocuite.

Nu există nici o asociere de taste pentru comanda replace-string. Totuși, puteți tasta orice comandă GNU Emacs, parcurgând următoarele etape:

1. Tastați M-x. Editorul afișează M-x în minibuffer și așteaptă să introduceți text.
2. Tastați comanda GNU Emacs și apăsați Enter. Pentru a folosi comanda replace-string, de exemplu, tastați **replace-string** și apoi apăsați Enter.
3. Pentru unele comenzi, GNU Emacs vă solicită să mai introduceți informații; introduceți-le. Când folosiți replace-string, de exemplu, editorul vă solicită mai întâi șirul ce trebuie înlocuit și apoi pe cel cu care trebuie înlocuit.

Când tastați o comandă GNU Emacs cu M-x, puteți apăsa tasta spațiu sau Tab pentru finalizarea execuției comenzii. Pentru a introduce comanda replace-string, de exemplu, ați putea începe tastând repl și apoi apăsând spacebar. Această acțiune determină GNU Emacs să afișeze replace- și apoi să facă o pauză. Tastați s și apăsați spacebar din nou. GNU Emacs afișează replace-string. Puteți folosi apoi comanda apăsând Enter. Încercați; veți vedea care este efectul.

Copierea și mutarea

Altă funcție uzuală pentru editare este copierea blocurilor de text și mutarea lor la altă locație din buffer. Primul pas în lucrul cu un bloc este definirea lui.



Observație

În editorul GNU Emacs, un bloc este definit ca fiind textul cuprins între un marcaj și poziția curentă a cursorului. Vă puteți gândi la acest *marcaj* ca la un indicator fizic plasat în buffer pentru a indica o locație. Pentru a fixa marcajul, deplasați cursorul la începutul blocului și apoi apăsați C-@ sau C-space (apăsați tasta Ctrl împreună cu tasta spațiu). GNU Emacs stabilește semnul la locația curentă și afișează următorul mesaj în minibuffer:

Mark set

După ce stabiliți marcajul, editorul tratează textul dintre locația curentă a cursorului și marcaj drept bloc. Pentru a copia blocul, tastați următoarele:

M-w

Această comandă copiază blocul într-o zonă de stocare internă, fără să-l șteargă din bufferul curent.

Dacă vreți să tăiați blocul de text, folosiți următoarea comandă:

C-w

Această comandă șterge blocul de text din buffer și-l mută într-o zonă de stocare internă.

Pentru a insera textul tăiat (sau copiat) în orice locație, deplasați cursorul în punctul de inserție și tastați următoarele:

C-y

Această comandă determină editorul GNU Emacs să lipească textul tăiat anterior în fața cursorului.

Între operațiile de tăiere și lipire, puteți comuta dintr-un buffer în altul și astfel să tăiați dintr-un fișier și să lipiți în altul. Pentru a tăia dintr-un fișier și a lipi în altul, parcurgeți următoarele etape:

1. Deschideți primul fișier folosind comanda C-x C-f. Pentru acest exercițiu, considerați că numele primului fișier este first.txt.
2. Deschideți al doilea fișier folosind comanda C-x C-f. Pentru acest exercițiu, considerați că numele celui de-al doilea fișier este second.txt.
3. Comutați în primul buffer folosind comanda C-x b first.txt
4. Deplasați cursorul la începutul textului ce trebuie copiat și apăsați C-space.
5. Deplasați cursorul la sfârșitul blocului și apăsați C-w pentru a decupa textul.
6. Mutați-vă în al doilea buffer folosind comanda C-x b second.txt.
7. Deplasați cursorul la locația unde vreți să inserați textul și apăsați C-y. GNU Emacs inserează textul tăiat anterior din primul buffer în cel de-al doilea.

Salvarea modificărilor

După ce modificați un buffer, trebuie să salvați acele modificări într-un fișier pentru a le face permanente. Comanda GNU Emacs pentru salvarea unui buffer în fișierul lui este `C-x C-s`. Această comandă salvează bufferul în fișierul cu același nume.

Pentru a salva bufferul într-un fișier cu nume diferit, folosiți comanda `C-x C-w`. Editorul GNU Emacs vă solicită să introduceți numele fișierului. Tastați numele și apăsați Enter pentru a salva bufferul în acel fișier. Spre deosebire de multe procesoare de text pentru DOS sau Windows, GNU Emacs nu adaugă automat o extensie fișierelor. De aceea, trebuie să introduceți numele întreg.

Rularea unui interpretor din editorul GNU Emacs

GNU Emacs posedă destul de multe facilități pentru a vă permite să accesați orice aplicație din Linux în cadrul unei sesiuni GNU Emacs. O modalitate de a accesa o aplicație din Linux este de a rula o sesiune de interpretor. Puteți face acest lucru folosind comanda GNU Emacs denumită `shell`. Pentru a vedea cum lucrează comanda `shell`, folosiți `M-x` (apăsați Escape, urmat de `x`). GNU Emacs afișează `M-x` în minibuffer și așteaptă să introduceți date în continuare. Tastați `shell` și apăsați Enter. GNU Emacs începe un nou proces interpretor și afișează promptul interpretor în fereastra GNU Emacs. Tastați orice comandă interpretor doriți. Rezultatul apare în fereastra GNU Emacs.

Puteți continua să lansați în execuție comenzi fiind în editor atât timp cât aveți nevoie. Toate ieșirile comenzilor sunt afișate în fereastra în care apare promptul interpretor. Când nu mai aveți nevoie de shell, tastați `exit`. Această comandă încheie procesul interpretor și vă readuce în GNU Emacs.

Dacă doriți să rulați o singură comandă interpretor, folosiți asocierea de taste `M-l`. Când folosiți `M-l`, editorul GNU Emacs afișează următorul prompt în minibuffer:

```
Shell command:
```

Tastați o comandă interpretor (precum `ls -l`) și apăsați Enter. GNU Emacs execută acea comandă și afișează rezultatul într-o fereastră separată. Pentru a reveni la modul de afișare cu o singură fereastră, tastați `C-x l`. Această comandă este Ctrl-x, urmată de numărul 1. Comanda instruește editorul să ștergă celelalte ferestre (în afară de cea care conține cursorul).

Scrierea paginilor man cu groff

Înainte de interfețele grafice, redactarea cu ajutorul calculatorului însemna pregătirea unui fișier text folosind comenzi de redactare încorporate și apoi procesarea celui fișier text formatat cu un program de calculator care genera comenzi pentru dispozitivele de ieșire: o imprimantă sau altă mașină de scris.

După cum știți, astfel de limbaje de formatare încă mai există. Un prim exemplu este Hypertext Markup Language (HTML), care este folosit pentru pregătirea paginilor World Wide Web.

La sfârșitul anilor '70 și începutul anilor '80, mi-am pregătit toată corespondența și rapoartele pe un sistem DEC VAX/VMS folosind un program numit RUNOFF. Acesta formata ieșirea pentru o imprimantă de linie sau o imprimantă cu rozetă. Acel program accepta comenzi încorporate, cum sunt următoarele:

```
.page size 58,80
.spacing 2
.no autojustify
```

După cum probabil v-ați dat seama, prima comandă setează numărul de linii pe pagină și numărul de caractere pe fiecare linie. A doua comandă generează o ieșire formatată la interval de două spații. Ultima comandă dezactivează alinierea. În esență, am introdus aceste comenzi într-un fișier text pe care l-am procesat prin intermediul aplicației RUNOFF și am trimis ieșirea la imprimanta de linie. Rezultatul arată la fel de bine ca un document tipărit la mașina de scris, care era destul de bun pentru majoritatea activităților din acea vreme.

UNIX a venit cu un program de redactare mai avansat denumit troff (care vine de la typesetting runoff) ce poate trimite rezultatul la un dispozitiv special numit typesetter. Un typesetter poate tipări mult mai bine decât o imprimantă de linie. Programul troff vă permite să alegeți diferite fonturi și să tipăriți textul cu caractere aldine (bold) sau înclinate (italic).

Pentru a gestiona ieșirea pe imprimante mai simple, UNIX a inclus de asemenea programul nroff (care vine de la nontypesetting runoff) pentru a procesa fișiere troff și a genera ieșiri, pentru a ignora comenzile de formatare mai speciale ce controlează aspectul documentului și a genera ieșiri pe o imprimantă de linie. Programul troff conține destul de multe facilități și, prin urmare, multe cărți de calculatoare au fost redactate cu acesta.

Acum, când aproape orice calculator are o interfață grafică (Microsoft Windows, Apple Macintosh sau X Window System), cele mai multe programe de procesare de text lucrează în modul „what you see is what you get – ceea ce vezi este ceea ce vei obține”, în care lucrați direct cu documentul formatat. Prin urmare, probabil că nu aveți nici un motiv pentru a folosi programul troff pentru redactare. Programul nroff este utilizat pentru o altă sarcină importantă: pregătirea paginilor man. Restul acestui capitol se concentrează asupra acestui aspect al utilizării programului nroff.



Observație

Programul groff este versiunea GNU a aplicațiilor troff și nroff. Cu indicatorii adecvați, puteți folosi groff pentru a așeza documentul în pagină pentru numeroase dispozitive de ieșire, inclusiv orice dispozitiv de tipul mașină de scris.

Chiar dacă nu folosiți groff pentru a pregăti documente formatare (deoarece folosirea unui procesor de texte destinat utilizării pe un PC este mai convenabilă), îl puteți utiliza pentru a scrie pagina man pentru orice program pe care-l realizați.

Paginile man sunt fișierele ce conțin informații pe care utilizatorii le pot vizualiza tastând comanda `man nume_program`. Această comandă prezintă informații de asistență online despre `nume_program`. Subiectul unei pagini man poate fi orice, de la o descriere generală a unui pachet software până la informații de programare despre o anumită funcție C (de exemplu, încercați `man fopen` pe sistemul dumneavoastră Linux).

După ce veți parcurge un exemplu, vă veți da seama că scrierea paginilor man cu programul groff este destul de simplă.



Observație

Înainte de a descrie procesul de pregătire al paginilor man, aș vrea să clarific faptul că nu se folosește programul groff pentru a pregăti pagina man. Pagina man este doar un fișier text conținând comenzi încorporate pe care groff le recunoaște. Puteți folosi groff pentru a vizualiza o pagină man în timpul pregătirii, dar puteți pregăti o pagină man fără să rulați groff.

Testarea unei pagini man existente

Înainte de a scrie o pagină man, ar trebui să examinați o pagină existentă. Un scurt exemplu este pagina man pentru aplicația nroff. Figura 24-5 prezintă pagina man a nroff într-o fereastră xterm.

Figura 24-5

Rezultatul
comenzii man
nroff într-o
fereastră xterm.

```

NAME
    nroff - emulate nroff command with groff

SYNOPSIS
    nroff [-h] [-i] [-name] [-trua] [-olist] [-ren]
          [-Tname] [file...]

DESCRIPTION
    The nroff script emulates the nroff command using groff.
    The -T option with an argument other than ascii and latin1
    will be ignored. The -h option is equivalent to the
    grotty -h option. The -i, -n, -s, -o and -r options have
    the effect described in troff(1). In addition nroff
    silently ignores options of -e, -q or -s.

SEE ALSO
    groff(1), troff(1), grotty(1)

Groff Version 1.11      26 June 1995      1
  
```

În acest caz, nu trebuie să fiți atent la conținutul exact al paginii man; ceea ce vă interesează este aranjarea. Examinați un moment așezarea în pagină. Veți observa următoarele lucruri:

- Numele comenzii apare în partea de sus a paginii man. Numărul 1 care apare între paranteze lângă numele comenzii semnifică secțiunea din manualul UNIX unde se găsește această comandă.
- Pagina man conține câteva secțiuni, fiecare secțiune fiind scrisă cu caractere aldine. Textul din interiorul secțiunii este decalat.
- În acest exemplu, secțiunile sunt NAME, SYNOPSIS, DESCRIPTION și SEE ALSO. Dacă examinați mai multe pagini man, veți observa că unele din ele au mult mai multe secțiuni. Totuși, aproape toate paginile man conțin aceste patru secțiuni.
- Unele porțiuni de text apar scrise cu caractere aldine.

Studierea unei pagini man sursă

După ce vizualizați o pagină man folosind comanda man, consultați fișierul sursă original din care comanda man generează ieșirea. În sistemul dumneavoastră Linux, fișierele sursă ale paginilor man se găsesc în câteva directoare. Numele acestor directoare sunt definite în fișierul `/etc/man.config`.

Fișierul sursă pentru pagina man `nroff` este `/usr/man/man1/nroff.1`. Examinați acest fișier pentru a vedea cum apare pagina man în forma finală, așa cum apare în figura 24-5. Iată cum puteți consulta fișierul sursă al paginii man pentru `gnroff`:

```

cd /usr/man/man1
more nroff.1
... Aici au fost sterse cateva linii.
.TH NROFF 1 "26 June 1995" "Groff Version 1.11"
.SH NAME
nroff \ - emuleaza comanda nroff cu groff
.SH SYNOPSIS
.B nroff
[
.B \ - h
]
[
.B \ - i
  
```

```

]
... Aici au fost sterse cateva linii.

[
.I file\ | . \ | . \ | .
]
.SH DESCRIPTION
scriptul
.B nroff
emuleaza comanda . B nroff
folosind groff .
Optiunea .B \ - T
cu un argument diferit de
.B ascii
sau
.B latin1
va fi ignorata.
... Aici au fost sterse cateva linii.
.SH "SEE ALSO"
.BR groff (1),
.BR gtroff (1),
.BR grotty (1)
  
```

O caracteristică interesantă a fișierului text formatat pentru pagina man este maniera întâmplătoare în care se rup liniile. Programul de formatare `groff` (comanda `man` folosește `groff` pentru a procesa fișierul text formatat) completează liniile de text, pentru ca documentul final să aibă un aspect prezentabil.

Majoritatea comenzilor `groff` trebuie să fie pe linii separate, fiecare comandă începând cu un punct. Puteți, totuși, să înglobați în text câteva comenzi `groff` pentru controlul fonturilor; aceste comenzi sunt precedate de caracterul backslash (`\`).

Iată un rezumat al comenzilor pe care le puteți găsi în fișierul sursă al paginii man `gnroff`:

- `.B` activează scrierea cu caractere aldine
- `.BI` activează scrierea cu caractere aldine și apoi comută pe scrierea cu caractere cursive
- `.BR` activează scrierea cu caractere aldine și apoi comută pe scrierea cu caractere romane
- `.I` activează scrierea cu caractere cursive
- `.SH` indică începutul unei secțiuni noi
- `.TH` indică titlul documentului

Dacă folosiți aceste comenzi cu punct (prin comenzi cu punct mă refer la acele comenzi care încep cu un punct) pentru a schimba fontul, fișierul sursă al paginii man tinde să aibă multe linii scurte, deoarece fiecare comandă cu punct trebuie să apară pe o linie separată. Așa cum o arată secțiunea următoare, puteți folosi comenzi înglobate pentru schimbarea fontului, ca să realizați un fișier sursă mai ușor de citit.

Scrierea unei pagini man demonstrative

Această secțiune vă arată cum să scrieți o pagină man pentru o aplicație cu caracter de exemplu numită `satview`. Să presupunem că programul `satview` afișează o imagine din satelit într-o fereastră. Programul are câteva opțiuni pentru specificarea proiecției hărții, procentului de transfocare și numelui fișierului ce conține datele despre imaginea prin satelit.

Folosiți un editor de text pentru a introduce codul sursă al paginii man prezentat în listingul următor și salvați-l într-un fișier numit `satview.1`.

```
. TH SATVIEW 1 "May 25, 1998" "Satview Version 4.01"
.SH NAME
satview \ - Vizualizeaza imagini din satelit.
.SH SYNOPSIS
\ fBsatview \ fP [ - p \ fIprojection \ fP ] [-m] [-z \ fIzoomlevel \ fP ]
\ fIfilename \ fP
.SH DESCRIPTION
\ fBsatview \ fP afiseaza imaginea din satelit din \ fIfilename \ fP.
.SS Options
.TP
\ fB - p \ fIprojection \ fR
Stabileste proiectia pe harta (poate fi \ fBL \ fR pentru Lambert Conformal sau
\ fBP \ fR pentru Polar Stereographic). Proiectia prestabilita este Lambert
Conformal.
.TP
\ fB - m \ fP
Include o harta in imaginea prin satelit..
.TP
\ fB - z \ fIzoomlevel \ fR
Stabileste scara de vizualizare (aceasta poate fi: 2, 4, sau 8 ). Scara de
vizualizare
prestabilita este 1.
.TP
\ fIfilename \ fR
Fisier ce contine date transmise prin satelit.
.SH FILES
.TP
\ fC /etc/awips/satview.rc \ fR
Comenzi de initializare pentru \ fBsatview \ fR
.SH SEE ALSO
nexrad (1), contour (1)
.SH Bugs
Pentru o scara de vizualizare mai mare ca 2, harta nu este aliniata corespunzator
cu imaginea.
```

Fișierul sursă are următoarele caracteristici:

- Eticheta `.TH` indică titlul paginii man, precum și data și un șir cu numărul versiunii. În pagina man formatată, șirurile cu data și versiunea apar în partea de jos a fiecărei pagini.
- Această pagină man are șase secțiuni – NAME, SYNOPSIS, DESCRIPTION, FILES, SEE ALSO și Bugs; fiecare dintre ele începe cu eticheta `.SH`.
- Secțiunea DESCRIPTION conține subsecțiunea Options. Eticheta `.SS` indică începutul unei subsecțiuni.
- În subsecțiunea Options, fiecare articol este listat cu eticheta `.TP`.
- Comanda `\fB` schimbă fontul în caractere aldine.
- Comanda `\fI` schimbă fontul în caractere cursive.
- Comanda `\fR` schimbă fontul în caractere romane.
- Comanda `\fP` reduce fontul în setarea precedentă.

Având aceste informații, puteți înțelege listingul codului sursă al paginii man.

Testarea și instalarea paginii man

Când pregătiți fișierul paginii man, doriți să vă asigurați că este formatat corect. Pentru a vizualiza rezultatul formatat, ar trebui să folosiți următoarea comandă:

```
groff -Tascii -man satview.1 | more
```

Această comandă rulează comanda `groff` cu dispozitivul `typesetter ascii` (ceea ce înseamnă că rezultatul este text ASCII simplu) și cu setul de macro-uri pentru pagini man (asta realizează opțiunea `-man`).

Dacă găsiți vreo discrepantă de formatare, verificați comenzile cu punct și orice comandă încorporată pentru schimbarea fontului și asigurați-vă că totul arată bine.

Dacă formatul paginii man vă satisface, puteți face pagina man disponibilă pentru toată lumea copiind fișierul într-unul din directoarele în care sunt localizate toate paginile man. Puteți instala pagina man cu următoarea comandă de copiere:

```
cp satview /usr/man/man1
```

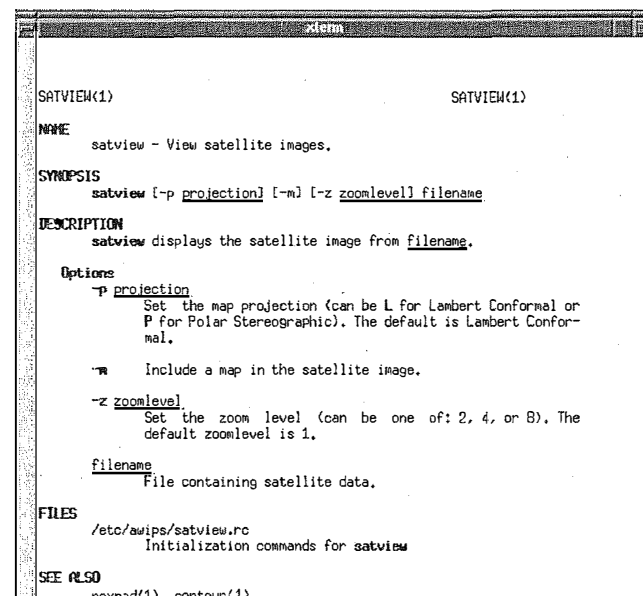
Apoi, încercați pagina man cu următoarea comandă:

```
man satview
```

Figura 24-6 vă prezintă rezultatul obținut. Comparați-l cu fișierul sursă din listingul 24-1 pentru a vedea efectele comenzilor în rezultatul final. Observați că comanda scrisă cu litere cursive generează un rezultat subliniat, deoarece fereastra terminalului nu poate afișa un font înclinat.

Figura 24-6

Rezultatul
comenzii man
satview într-o
fereastră xterm.



Cu listingul din această secțiune drept ghid, vă puteți redacta propriile dumneavoastră pagini man.

Rezumat

Configurarea multor caracteristici Linux depinde de setările stocate în fișiere text. Prin urmare, trebuie să știți cum să editați aceste fișiere pentru a seta și întreține un sistem Linux. Altă formă de procesare de text implică redactarea și formatarea documentelor text. Linux conține programul `groff` în acest scop. În prezent, probabil că dumneavoastră folosiți un PC pentru a pregăti documente formate, dar încă mai aveți nevoie să pregătiți un tip de document formatat: paginile man, care furnizează asistență online despre orice

produs software pe care-l proiectați. Ca urmare, acest capitol vă arată cum să folosiți câteva editoare de text și cum să folosiți paginile man folosind programul `groff`. Citind acest capitol, învățați următoarele:

- ▶ Linux conține o varietate de editoare de text. Este important să învățați editoarele `ed` și `vi`, deoarece toate sistemele UNIX conțin aceste editoare. Cu toate că editorul `ed` este criptic, uneori va trebui să editați fișiere și-l veți folosi deoarece nu aveți alt editor disponibil.
- ▶ Editorul `ed` lucrează cu un fișier text citind linie cu linie (de aceea este denumit editor orientat spre linie). Fiecare comandă ia forma unui grup de linii, urmată de o comandă formată dintr-un caracter ce se aplică acelui grup de linii.
- ▶ `vi` este un editor pe toată pagina ce vă permite să vizualizați simultan mai multe linii dintr-un fișier. Editorul `vi` are trei moduri: modul comandă vizuală, modul comandă cu două puncte (:) și modul introducere text.
- ▶ În modul introducere text puteți introduce text. Pentru a comuta în acest mod, tastați `i`, `a` sau `A`. Pentru a reveni în modul comandă apăsați `Escape`.
- ▶ În modul comandă cu două puncte a editorului `vi`, introduceți semnul două puncte (:) urmat de o comandă ce are aceeași sintaxă ca și comenzile din editorul `ed`.
- ▶ GNU Emacs este nu numai un editor, ci și un mediu în care puteți executa multe operații de rutină.
- ▶ Ca editor, GNU Emacs nu are nici un mod; orice tastați este introdus direct într-un fișier. Comenzile încep cu un caracter de control (precum `Ctrl-x`) sau `Escape`.
- ▶ Pentru a vedea îndrumarul online, tastați `Ctrl-h t` în GNU Emacs.
- ▶ Utilitarul `groff` este folosit pentru formatarea documentelor. `groff` asigură funcționarea utilitatelor standard din UNIX, `nroff` și `troff`.
- ▶ Chiar dacă nu pregătiți documente formate în Linux (deoarece această operație este mai simplă pe un PC sau un Macintosh), aveți nevoie în continuare de `groff` pentru pregătirea paginilor man.
- ▶ Modalitatea cea mai indicată de a învăța să pregătiți o pagină man este să studiați o pagină deja existentă și să copiați stilul în propria dumneavoastră pagină man. Acest capitol vă oferă un exemplu din care puteți învăța cum să pregătiți o pagină man tipică.

ANEXE

- ▶ **Anexa A:** Breviar de aplicații Linux
- ▶ **Anexa B:** Comenzi Linux
- ▶ **Anexa C:** Resurse Linux
- ▶ **Anexa D:** Despre CD-ROM

Anexa A

Breviar de aplicații Linux

Versiunea Linux Red Hat de pe CD-ROM-ul anexat acestei cărți conține multe aplicații. Mult mai multe aplicații Linux sunt disponibile pe Internet. Acest capitol descrie pe scurt multe din aplicațiile Linux de pe CD-ROM-ul anexat. Scopul anexe de față este să vă prezinte câteva din aplicații, pentru a ști că sunt disponibile pe CD-ROM.

Iată un set de aplicații Linux selectat de pe CD-ROM, organizat pe categorii:

Categorie	Aplicații
Editoare	GNU Emacs, JED, Joe, vim
Utilitare	DOSEMU, GNU bc, gzip, ispell, Midnight Commander, patch, xfm
Grafică și imagini	XV, XPaint, Xfig, Gnuplot, Ghostscript, Ghostview, GIMP, ImageMagick, xanim
Internet	dip, elm, imap, ircii, inn, pine, pppd, rdist, sendmail, tin, trn, x3270, Netscape Communicator
Comunicații seriale	Minicom, Seyon
Programare	Bison, f2c, flex, Gawk, GCC, p2c, Perl, Python, RCS, Tcl/Tk, yacc
Formatare text	groff, TeX
Jocuri	GNU Chess, xtetris, acm, colour-yahtzee, flying, fortune-mod, mysterious, paradise, xchomp, xevil, xgalaga, xgammon, xpilot

Această listă nu este completă. Linux include o multitudine de comenzi standard din UNIX, de exemplu, care nu apar în listă.

Acest capitol vă pune la dispoziție un scurt rezumat al unui mic subset din aceste aplicații. În particular, voi pune accentul pe câteva programe utilitare ce nu au fost tratate în capitolele anterioare. Unele aplicații, cum ar fi GNU Emacs și DOSEMU, sunt descrise detaliat în capitolele anterioare ale acestei cărți. De asemenea, categorii întregi de aplicații importante, cum ar fi cele pentru Internet, sunt descrise cu lux de amănunte în capitole individuale.

Doar câteva din aplicațiile listate sunt instalate automat o dată cu procesul de instalare Linux. La câteva aplicații veți întâmpina probleme la instalarea de pe CD-ROM.

Remarcați că unele din aceste aplicații sunt shareware. Dacă aveți de gând să le folosiți în munca dumneavoastră trebuie să vă înregistrați și să plătiți o taxă shareware. Această informație este menționată acolo unde este necesar.

Editoare

Pe CD-ROM-ul anexat puteți găsi următoarele editoare de text:

- GNU Emacs, singurul Emacs original
- JED, un editor asemănător cu GNU Emacs
- Joe, un editor de text ce are comenzi asemănătoare cu cele din WordStar.
- Vim, un editor asemănător cu vi.

Așa cum puteți vedea, majoritatea acestor editoare sunt similare cu GNU Emacs sau vi. Aceste editoare care emulează alte programe similare sunt de obicei mai mici și au mai puține facilități.

GNU Emacs

CD-ROM-ul anexat acestei cărți conține GNU Emacs Versiunea 20.2.1. GNU Emacs, care este unul dintre cele mai cunoscute produse GNU, este distribuit de Free Software Foundation. CD-ROM-ul conține două versiuni ale acestui editor:

- Versiunea completă GNU Emacs, care lucrează sub x și dispune de meniuri pentru operațiile de editare. Trebuie să aveți instalat x pentru a rula această versiune. Fișierul binar este configurat cu următoarele opțiuni:

```
--with-x11 --with-x-toolkit --prefix=/usr
```

- O versiune cu aproximativ 330 KB mai mică decât versiunea completă GNU Emacs, cu aceleași facilități, dar fără suport pentru x.

Puteți rula versiunea non-X a editorului GNU Emacs într-o fereastră xterm sub X. Totuși, versiunea non-X este special destinată acelor utilizatori care nu au instalat X Window System pe sistemele lor.

Capitolul 24 descrie caracteristicile editorului GNU Emacs și arată cum trebuie folosit pentru editare de text.

JED

Ca și alte editoare disponibile gratuit, acest editor a împrumutat numele creatorului său, John E. Davis. JED este un editor puternic dar mic, asemănător cu GNU Emacs, care la fel ca și GNU Emacs suportă moduri de editare și poate citi fișiere info. JED poate, de asemenea, emula alte editoare de text, precum WordStar și editorul EDT al sistemelor VAX/VMS. Una din trăsăturile unice ale editorului JED este posibilitatea de a edita cu el fișiere binare.

Veți găsi fișierul RPM pentru JED în directorul RedHat/RPMS de pe CD-ROM-ul însoțitor.

Joe

CD-ROM-ul anexat conține editorul de text Joe versiunea 2.2. Joe este denumit după creatorul său, Joseph H. Allen. Editorul Joe folosește comenzi asemănătoare cu cele ale popularului procesor de texte din perioada anterioară, WordStar.

Veți găsi fișierul RPM pentru Joe în directorul RedHat/RPMS de pe CD-ROM.

Vim

Vim vine de la Vi *IM*proved. Așa cum arată numele, Vim este versiunea îmbunătățită a editorului de text standard din UNIX vi. CD-ROM-ul conține versiunea 3.0 a editorului Vim. Pe lângă comenzile vi standard, Vim include câteva caracteristici noi, cum ar fi câteva niveluri de anulare a comenzilor, istoricul liniei de comenzi și completarea numelor fișierelor.



Referință încrucișată

Capitolul 24 vă arată cum să folosiți editorul vi. Dacă cunoașteți vi, puteți folosi Vim cu ușurință.

Utilitare

Această categorie cuprinde multe aplicații (folositoare pentru realizarea a diverse operații) livrate o dată cu Linux. Lista nu este foarte cuprinzătoare, deoarece am inclus numai aplicațiile cele mai populare. Aplicațiile descrise pe scurt în următoarele secțiuni sunt:

- DOSEMU, un emulator DOS pentru Linux
- GNU bc, un calculator cu precizie arbitrară.
- gzip, un utilitar pentru comprimarea și expandarea fișierelor
- ical, un program calendar bazat pe X
- ispell, un verficator ortografic interactiv pe tot ecranul
- Midnight Commander, un program de gestiune a fișierelor cu o interfață bazată pe text
- patch, un utilitar pentru aplicarea modificărilor unui fișier text
- xfm, un program de gestiune a fișierelor bazat pe X

DOSEMU

DOSEMU este un emulator DOS distribuit gratuit – un program ce poate fi rulat sub Linux pentru a obține un mediu DOS. Cu acest program puteți rula aplicații DOS sub Linux. DOSEMU nu este instalat prestabilit. Puteți găsi programul DOSEMU în directorul contrib de pe CD-ROM.



Referință încrucișată

Pentru a folosi aplicația DOSEMU, trebuie mai întâi să o instalați de pe CD-ROM. Capitolul 7 vă arată cum să instalați și să folosiți acest program.

GNU bc

GNU bc vă permite să introduceți numere cu precizie arbitrară și să realizați cu acestea diferite calcule. GNU bc implementează facilitatea de calcul cu precizie arbitrară specificată de standardul preliminar POSIX P1003.2/D11.

Aveți șansa să instalați GNU bc ca parte a unui pachet în timp ce instalați Linux de pe CD-ROM-ul anexat acestei cărți.

Dacă aveți instalat GNU bc, îl puteți rula cu următoarea comandă:

```
bc
```

```
bc 1.04
```

Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation,

- -S .xxx determină gzip să folosească sufixul .xxx în loc de .gz
- -t testează integritatea unui fișier comprimat
- -v afișează numele și procentul de comprimare pentru fiecare fișier
- -V afișează numărul versiunii și opțiunile compilatorului folosite pentru a construi acea versiune de gzip
- -l utilizează cea mai rapidă metodă de compresie (chiar dacă nu este foarte eficientă)
- -9 folosește cea mai lentă metodă de compresie (comprimarea este cea mai eficientă)
- file... reprezintă unul sau mai multe nume de fișiere (ce trebuie comprimate sau decomprimate)

Utilitarul gzip este instalat automat atunci când instalați Linux de pe CD-ROM.

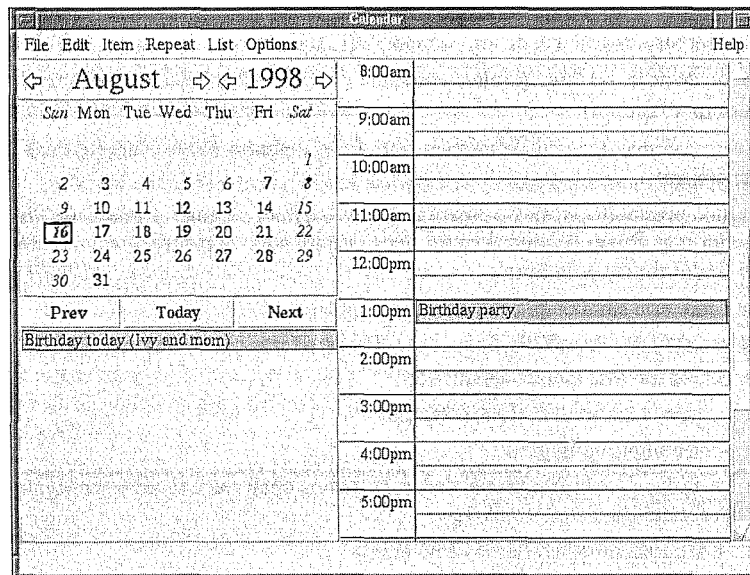
ical

Utilitarul ical este un program calendar interactiv ce are la bază sistemul X Window System. Când instalați Linux de pe CD-ROM, instalați de asemenea și ical.

Pentru a folosi ical, tastați ical dintr-o fereastră xterm sau selectați din meniu Start->Programs->Utilities->Ical. Programul afișează o fereastră pe tot ecranul în care puteți executa clic pe o anumită dată calendaristică pentru a vizualiza programul acelei zile. Figura A-1 prezintă un calendar tipic.

Figura A-1

Vizualizarea
calendarului cu
aplicația ical



Vă puteți deplasa la altă lună sau la alt an executând clic pe săgețile de lângă lună sau an. Pentru o anumită dată calendaristică, puteți adăuga date în zona de sub calendarul lunii.

Pentru a adăuga întâlniri la anumite ore, executați clic pe ora respectivă și tastați o scurtă descriere a întâlnirii.

După ce terminați de adăugat evenimentele și întâlnirile, selectați Save din meniul File pentru a salva calendarul.

ispell

Utilitarul ispell este un verficator ortografic interactiv. Când instalați Linux de pe CD-ROM-ul anexat acestei cărți, aveți șansa să instalați și ispell.

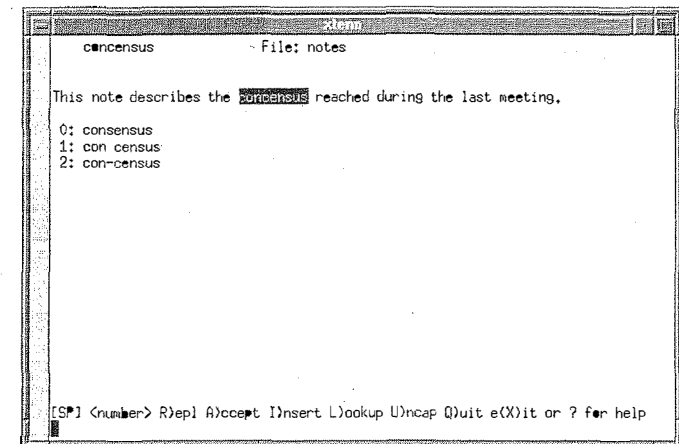
Utilizarea utilitarului ispell este simplă. De cele mai multe ori, folosiți ispell pentru verificarea ortografică a cuvintelor dintr-un fișier text. Pentru aceasta, pur și simplu tastați **ispell filename**. Să presupunem că ați introdus câteva rânduri și le-ați stocat într-un fișier text numit notes. Pentru verificarea ortografică a acestui fișier, tastați comanda:

```
ispell notes
```

După ce rulează, ispell examinează fișierul denumit notes până când găsește un cuvânt scris greșit (sau orice cuvânt care nu se găsește în dicționarul programului). În fișierul notes, ispell poate găsi cuvântul scris greșit consensus. Figura A-2 vă arată raportul întocmit de ispell la întâlnirea acestei greșeli de ortografie într-o fereastră text extinsă pe tot ecranul.

Figura A-2

Un cuvânt scris
greșit și
posibilitățile
sugerate de
ispell



Așa cum se vede în figura A-2, ispell vă arată cuvântul scris greșit evidențiat în propoziția în care apare. Sub acea propoziție, ispell prezintă corecturile posibile. În acest caz, utilitarul listează consensus – alegerea corectă – ca primă corectură pentru consensus.

În partea de jos a ecranului, ispell afișează comenzile pe care le introduceți. Iată care sunt semnificațiile articolelor de pe ultima linie din figura A-2:

- Space înseamnă că acceptă cuvântul
- <number> se referă la numărul corecției, dintre cele sugerate de utilitar, pe care doriți s-o folosiți. Pentru a înlocui cuvântul scris greșit cu corecția numărul 0, pur și simplu tastați 0.
- R înseamnă înlocuire. ispell solicită un cuvânt de înlocuire.
- A înseamnă acceptare. ispell acceptă cuvântul pentru restul fișierului.
- I înseamnă inserare. ispell acceptă cuvântul și-l adaugă, de asemenea, la dicționarul personal al utilizatorului.
- L determină ispell să caute cuvinte în dicționarul sistem.
- U înseamnă că acceptă cuvântul și adaugă o versiune cu litere mici în dicționarul personal al utilizatorului.

- Q determină ispell să-și încheie imediat execuția.
- X determină ispell să salveze restul fișierului, ignorând greșelile de ortografie.
- ? afișează un ecran de asistență

În plus, puteți introduce o comandă de interpretor având ca prefix semnul exclamării (!). Dacă imaginea pe ecran devine neclară dintr-un motiv oarecare, apăsați Ctrl-L pentru a o reîmprospăta.

Midnight Commander

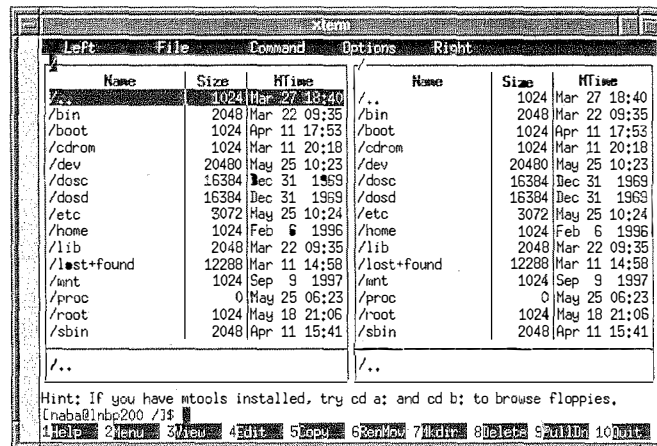
Midnight Commander este un program de gestiune a fișierelor cu o interfață bazată pe text, asemănătoare cu Norton Commander pentru MS-DOS. Puteți folosi Midnight Commander pentru a gestiona fișiere și directoare, fără să fie necesar să învățați comenzi UNIX. Dacă nu sunteți sigur de comenzile UNIX, veți dori probabil să folosiți Midnight Commander pentru a efectua operații cu fișiere precum copiere, redenumire și ștergere. Puteți chiar să editați și să vizualizați un fișier cu Midnight Commander.

Când instalați Linux de pe CD-ROM, aveți posibilitatea să instalați și Midnight Commander. Puteți rula Midnight Commander de la consolă sau dintr-o fereastră *xterm*. Pentru a porni Midnight Commander, tastați *mc*.

Figura A-3 vă arată un ecran tipic Midnight Commander. În acest caz, directorul curent s-a întâmpnat să fie directorul rădăcină (/).

Figura A-3

Ecranul Midnight Commander într-o fereastră *xterm*



Ecranul afișat de Midnight Commander conține două liste de directoare alăturate. Puteți schimba directorul în oricare din liste. Cele două imagini alăturate vă permit să efectuați operații precum copierea sau mutarea între două directoare.

În partea de jos a ecranului Midnight Commander sunt zece comenzi care seamănă vag cu butoanele dintr-o interfață grafică. Aceste comenzi sunt operațiile uzuale pe care le puteți efectua în fereastra Midnight Commander.

Midnight Commander răspunde la acțiunile mouse-ului într-o fereastră consolă precum și într-o fereastră *xterm*. Aceasta înseamnă că puteți executa dublu clic pe numele unui director pentru a-i vedea conținutul în ecranul afișat de Midnight Commander. De asemenea, puteți

executa clic pe comenzi (afișate în video invers în partea de jos a ecranului Midnight Commander) pentru a le activa.

patch

Utilitarul GNU *patch* este destinat aplicării remediilor (*patches*) în fișiere. Ideea de bază a acestui utilitar este aceea că, atunci când doriți să distribuiți modificările pentru un fișier, rulați comanda standard *diff* din UNIX și generați un fișier *diff* care arată cum trebuie modificat fișierul. Apoi, distribuiți acel fișier *diff* fiecărei persoane care are fișierul original. Destinatul rulează utilitarul *patch* având ca argument de intrare fișierul *diff*, iar *patch* face modificările în fișierele originale.



Referință încrucișată

Utilitarul *patch* este instalat automat atunci când instalați Linux de pe CD-ROM. Capitolul 2 vă arată cum să folosiți acest program pentru a aplica modificările surselor nucleului Linux, atunci când îmbunătățiți nucleul de la o versiune la următoarea.

Puteți învăța mai multe despre *patch* printr-un exemplu simplu. Să presupunem că aveți un fișier denumit *original.txt*, ce conține următorul text:

```
Version: 1.0
```

Revision history:

```
5/25/98: Original file (NB)
```

– Acest fișier text este folosit ca exemplu pentru ilustrarea modului de folosire a comenzilor *diff* și *patch* pentru actualizarea unui fișier.

Să presupunem că ați distribuit deja acest fișier la câțiva utilizatori. (Să presupunem că acest fișier este fișierul sursă al unui program de calculator.) După ceva timp, faceți câteva modificări în acest fișier. Noul fișier, denumit *revised.txt*, arată astfel:

```
Version: 1.1
```

Revision history:

```
5/25/98: Original file (NB)
```

```
6/15/98: Added a new line (LB)
```

Acest fișier text este folosit ca exemplu pentru ilustrarea modului de folosire a comenzilor *diff* și *patch* pentru actualizarea unui fișier.

Ceva nou adăugat . . .

Acum doriți să distribuiți aceste modificări utilizatorilor, astfel încât ei să poată folosi noul fișier. Prima dumneavoastră sarcină este să creați un fișier *diff*, care capturează modificările operate.

Pentru a crea acest fișier, rulați *diff* cu opțiunea *-u* și specificați cele două nume de fișiere ca argumente – fișierul original urmat de cel revizuit. Astfel, următoarea comandă creează fișierul *diff* pentru exemplul curent:

```
diff -u original.txt revised.txt > patch-1.1
```

Această comandă creează fișierul *patch-1.1*, pe care ar trebui să-l distribuiți utilizatorilor ce folosesc curent fișierul *original.txt*. Acest fișier *diff* se mai numește și fișier *patch*.

Când utilizatorii primesc fișierul *patch* (fișierul de remedii), trebuie doar să-l plaseze în același director cu fișierul *original.txt* și să tasteze următoarea comandă:

```
patch < patch-1.1
```

```
Hmm . . . Looks like a unified diff to me . . .
```

The text leading up to this was:

```

| --- original.txt      Mon May 25 11:27:24 1998
| +++ Revised.txt      Mon Jun 15 11:30:00 1998

```

Patching file original.txt using Plan A . . .

Hunk #1 succeeded at 1.

Done

Utilitarul patch raportează unele mesaje folositoare și aplică modificările din fișierul patch. Mai întâi, patch copiază fișierul original.txt în fișierul original.txt.orig; apoi aplică modificările direct fișierului original.txt. După ce patch își încheie execuția, veți observa că informațiile conținute în fișierul original.txt se potrivesc cu cele din revised.txt. Acest exemplu reprezintă un model de folosire a utilitarului patch pentru actualizarea fișierelor text.

xfm

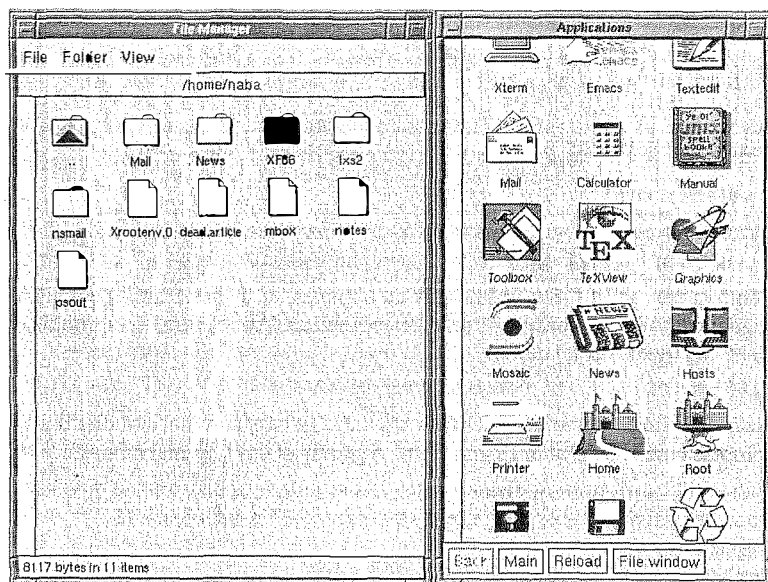
Programul xfm este un fișier bazat pe sistemul X și un utilitar ce gestionează aplicațiile. Când instalați prima dată Linux de pe CD-ROM-ul anexat acestei cărți, puteți opta pentru a instala xfm.

Înainte de a folosi xfm, rulați un fișier script — /usr/X11R6/bin/xfm.install — care instalează unele fișiere de configurare prestabilite în subdirectorul .xfm al directorului dumneavoastră de bază (fișierul script creează un director .xfm, dacă acesta nu există). Deoarece directorul /usr/X11R6/bin se află de obicei în variabila dumneavoastră de mediu PATH, puteți rula scriptul de instalare xfm cu următoarea comandă:

```
xfm.install
Default configuration files installed. (Fișierele de configurare prestabilite
sunt instalate.)
```

După parcurgerea acestei etape, puteți porni aplicația xfm într-o fereastră xterm tastând **xfm&**. Figura A-4 prezintă setarea inițială a ferestrelor xfm.

Figura A-4
Ferestrele inițiale
ale programului
xfm



Fereastra File Manager afișează directoarele și fișierele din directorul de bază al utilizatorului. Directoarele apar ca dosare (folders) — o metaforă populară folosită de multe programe de gestiune a fișierelor, inclusiv cele recente, cum ar fi Explorer din Microsoft Windows 95.

Fereastra Applications afișează un set de pictograme ce reprezintă aplicațiile. Conținutul ferestrei depinde de un fișier pentru configurarea textului denumit Apps (localizat în subdirectorul .xfm al directorului dumneavoastră de bază). Fișierul prestabilit Apps conține următorul cod:

```

#XFM
Xterm : : : terminal . xpm : exec xterm :
Emacs : : : emacs . xpm : exec emacs : exec emacs $*
Textedit : : : edit . xpm : exec textedit : exec textedit $*
Mail : : : mailtool . xpm : exec xmailtool :
Calculator : : : calc . xpm : exec xcalc :
Manual : : : man2 . xpm : exec xman :
Toolbox : - / . xfm : Toolbox : tools . xpm : LOAD :
TeXView : : : tex - view . xpm : : exec xdvi $1
Graphics : - / . xfm : Graphics : drawing . xpm : LOAD :
Mosaic : : : xmosaic . xpm : exec mosaic :
News : : : news . xpm : exec xrn :
Hosts : - / . xfm : Hosts : rlogin . xpm : LOAD :
Printer : : : printer . xpm : lpq -Plaser : exec lpr -Plaser $*
Home : : : citadel . xpm : OPEN :
Root : / : : citadelroot . xpm : OPEN :
A \ : : / disk : a : floppy . xpm : OPEN :
B \ : : / disk : b : disk . xpm : OPEN :
Trash : : : trash : recycle . xpm : OPEN : shift ; mv -f -b -V numbered $* ~ / .
trash

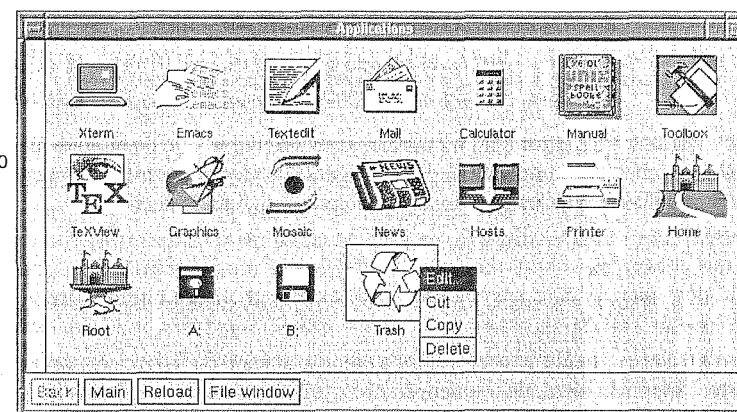
```

Din acest listing puteți intuit sintaxa fișierului Apps. Liniile care încep cu semnul diez (#) reprezintă comentarii. Alte linii prezintă o etichetă, un fișier cu pictograma formată dintr-o hartă de pixeli (cu extensia .xpm) și o comandă ce va fi executată când utilizatorul execută dublu-clic pe acea pictogramă.

Nu este nevoie să învățați sintaxa fișierului Apps, deoarece îi puteți edita conținutul direct din fereastra Applications. Când executați clic cu butonul drept al mouse-ului oriunde în fereastra Applications, apare un meniu pop-up sensibil la context. Dacă executați clic cu butonul drept pe o pictogramă, xfm afișează meniul pop-up prezentat în figura A-5.

Figura A-5

Acest meniu
pop-up apare
atunci când
executați clic
cu butonul drept
pe o pictogramă
din fereastra
Applications.



Meniul pop-up vă permite să efectuați una din cele câteva operații care se află pe pictogramă. Dacă executați clic cu butonul drept pe pictograma Trash și apoi alegeți Edit din meniul pop-up, de exemplu, xfm afișează altă casetă de dialog, în care puteți modifica acțiunea realizată de pictograma Trash.

Pentru a adăuga o nouă aplicație în fereastra Applications, executați clic cu butonul drept într-un spațiu vid din fereastră. Figura A-6 vă prezintă meniul pop-up care apare.

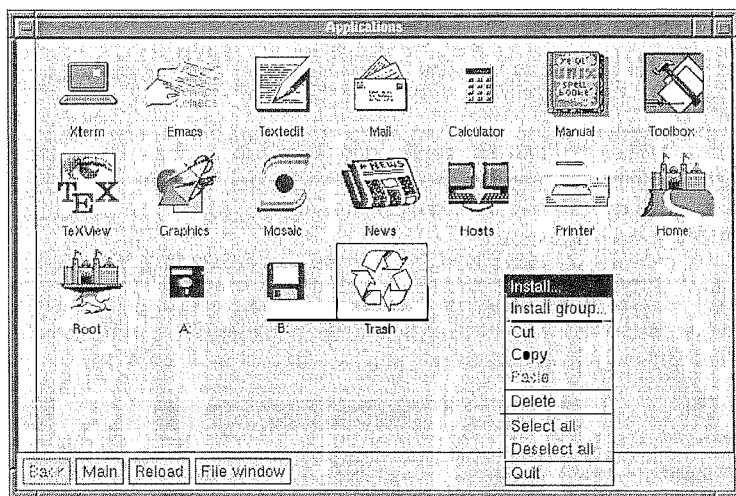
Așa cum o arată figura A-6, meniul pop-up vă permite să adăugați sau să eliminați aplicații. Dacă selectați Install din meniul pop-up, xfm afișează o casetă de dialog în care puteți completa informațiile solicitate și apoi executați clic pe butonul Install.

Trebuie să executați dublu-clic pe o pictogramă din fereastra Applications pentru a lansa aplicația asociată.

În fereastra File Manager, xfm nu arată, în mod prestabilit, nici un fișier sau director al cărui nume începe cu un punct (.). Prin convenție, aceste fișiere sunt considerate ascunse în UNIX. Pentru a vedea aceste fișiere ascunse, executați clic pe eticheta View de pe marginea superioară a ferestrei (bara de meniu). Din meniul pop-up view care apare, selectați Show hidden files. Apoi, xfm afișează toate fișierele și directoarele, inclusiv cele ale căror nume încep cu un punct (.).

Figura A-6

Acest meniul pop-up apare atunci când executați clic cu butonul drept într-un spațiu vid din fereastra Applications.



Pentru a vizualiza conținutul unui director, executați dublu-clic pe numele lui. Fereastra File Manager vă arată conținutul acelui director. Pentru a vă deplasa înapoi în directorul părinte, executați dublu-clic pe dosarul care are două puncte (..) drept etichetă.

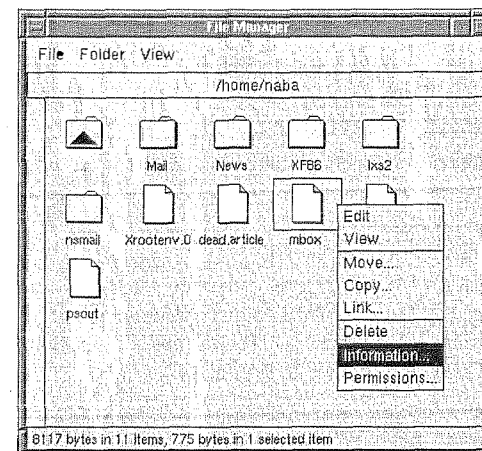
Fereastra File Manager dispune, de asemenea, de meniuri desfășurabile contextuale care apar atunci când executați clic cu butonul drept pe o pictogramă. Figura A-7 vă arată meniul desfășurabil care apare atunci când executați clic cu butonul drept pe pictograma unui fișier.

Puteți folosi acest meniul pop-up pentru a efectua operații cu fișiere precum copiere, mutare și ștergere. Puteți vizualiza, de asemenea, informații detaliate despre fișier și puteți modifica setările privind accesul la acesta. Pentru a vedea informații despre fișier, de exemplu, selectați Information din meniul pop-up. Apare o casetă de dialog cu informații despre fișierul pe care

ati executat clic cu butonul drept. După ce vizualizați informațiile despre fișier, executați clic pe butonul Ok pentru a elimina de pe ecran această casetă de dialog.

Figura A-7

Acest meniul pop-up apare atunci când executați clic cu butonul drept pe o pictogramă din fereastra File Manager.



În final, pentru a părăsi xfm, selectați Quit din meniul File al ferestrei File Manager. Când xfm solicită confirmarea, executați clic pe butonul Continue (dacă chiar vreți să părăsiți programul).

Această scurtă prezentare nu a descris în totalitate toate facilitățile aplicației xfm. Puteți, de exemplu, să configurați fereastra Applications cu aplicațiile dumneavoastră favorite trăgând și plasând pictogramele aplicațiilor și fișierelor script în fereastră. Modalitatea cea mai bună pentru a învăța mai mult despre xfm este să începeți să-l folosiți. Dacă aveți de gând să folosiți un administrator de fișiere, ar trebui să încercați xfm.

Grafică și imagini

Aplicațiile din această categorie vă permit să pregătiți, să vizualizați, să modificați și să tipăriți grafică și imagini. Următoarele aplicații sunt rezumate în secțiunile care vor urma:

- XV, un program shareware pentru vizualizarea și manipularea multor tipuri de imagini
- XPaint, un program de desenare în format hartă de biți realizat având ca model MacPaint
- Xfig, un program pentru realizarea desenelor tehnice
- Gnuplot, un pachet pentru plotare
- Ghostscript, un interpretor PostScript capabil de a realiza ieșiri pe multe dispozitive, inclusiv în diferite formate de fișiere imagini
- Ghostview, o aplicație X ce servește drept front end pentru Ghostscript

XV

xv este un program shareware impresionant pentru vizualizarea a aproape oricărui tip de imagine (GIF, JPEG, PostScript, Windows BMP, TIFF, PCX și X11 Bitmap File, pentru a numi numai câteva) pe un ecran X. Dacă sunteți interesat de folosirea programului, ar trebui să citiți informațiile referitoare la licența de utilizare, care apar în documentația aferentă.

Când instalați Linux de pe CD-ROM, puteți alege să instalați programul XV; el se află în pachetul Graphics Manipulation.

Dacă programul XV este instalat, îi veți găsi documentația în directorul `/usr/doc/xv-3.10a`. În particular, ar trebui să găsiți fișierul PostScript `/usr/doc/xv-3.10a/xvdocs.ps` cu mărimea de aproape 4 MB. El reprezintă manualul pentru programul XV.

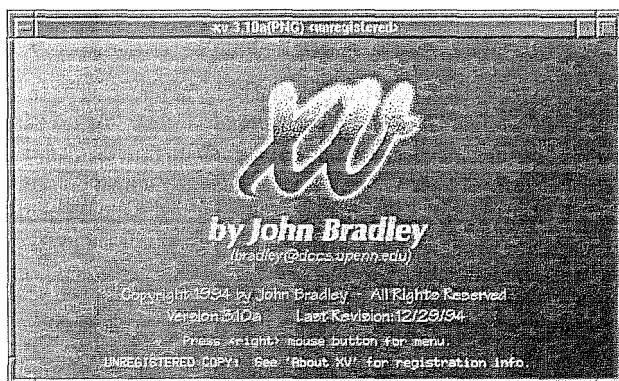
XV are atât de multe caracteristici, încât o scurtă descriere nu vă poate oferi o imagine corespunzătoare a posibilităților acestuia. Ar trebui să găsiți o imprimantă PostScript și să tipăriți fișierul `xvdocs.ps`. Documentația are aproape 130 de pagini și conține multe imagini, de aceea tipărirea s-ar putea să dureze ceva timp.

În documentație veți găsi de asemenea, licența de utilizare a acestui program. Ar trebui să o citiți pentru a vedea ce aveți de făcut dacă aveți de gând să folosiți XV în munca dumneavoastră. Deoarece programul XV este un produs shareware, autorul vă încurajează să-l folosiți și chiar să-l distribuiți altor utilizatori.

Pentru a încerca programul XV, tastează `xv&` într-o fereastră `xterm` (trebuie să rulați `x` pentru a putea folosi acest program). Din suprafața de lucru a ferestrei X puteți de asemenea să selectați din meniu `Start->Programs->Graphics->xv`. Programul XV pornește afișându-și emblema într-o fereastră inițială, așa cum se vede în figura A-8.

Figura A-8

Fereastra inițială a programului XV



Bara de titlu a ferestrei indică numărul versiunii programului XV și faptul că această copie nu este înregistrată. Textul din fereastră vă indică să executați clic cu butonul drept pentru a afișa meniul. Dacă executați clic cu butonul drept în fereastră, este adusă în prim-plan fereastra de control a programului XV, indicată în figura A-9.

Așa cum sugerează și numele, fereastra de control XV este centrul de control al programului; toate operațiile le executați din această fereastră.

Dacă nu ați citit încă licența, executați clic pe butonul `About xv` din fereastra de control XV. Programul deschide o fereastră pentru vizualizarea textului ce afișează informații referitoare la licența XV.

O operație uzuală este încărcarea unei imagini. Pentru a face acest lucru, executați clic pe butonul `Load` din fereastra de control. Programul XV afișează o casetă de dialog pentru selectarea fișierului, ca în figura A-10.

Prin intermediul casetei de dialog din figura A-10, puteți selecta un fișier imagine dintr-un director. În Figura A-10, este selectat fișierul `linux-penguin.gif`. După extensia `.gif`, vă dați seama că este vorba despre un fișier GIF, care poate fi vizualizat cu programul XV.

Figura A-9

Fereastra de control a programului XV

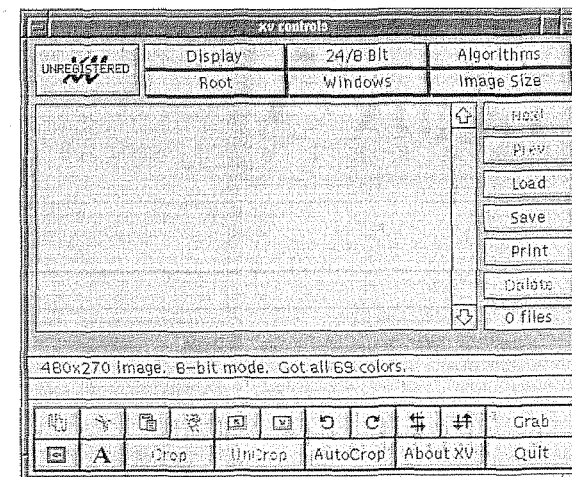
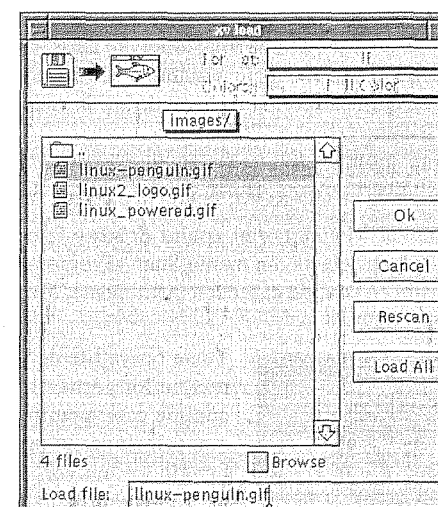


Figura A-10

Selectarea unui fișier imagine ce va fi vizualizat cu programul XV



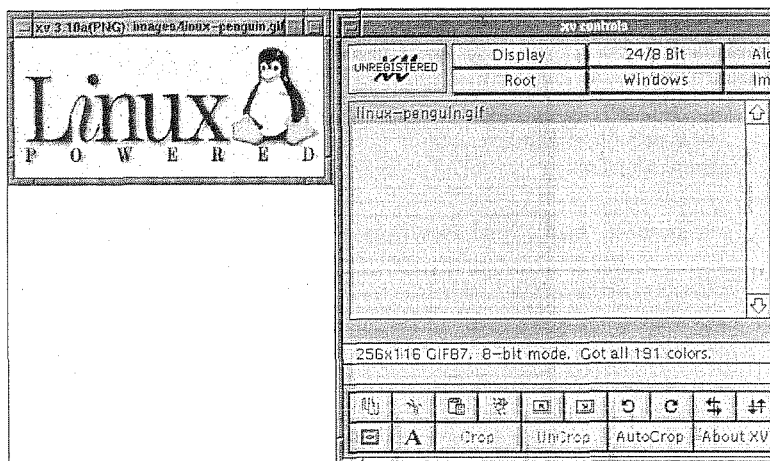
Pentru a vizualiza fișierul imagine selectat, executați clic pe butonul `Ok`. Această acțiune determină programul să încarce imaginea în fereastra imagine și să afișeze numele fișierului în fereastra de control a programului XV, așa cum se vede în figura A-11.

Fereastra de control din XV vă va prezenta câteva informații detaliate despre imagine, care este în formatul GIF87 și are 191 de culori. Mărimea imaginii este de 256x116 pixeli.

Aveți posibilități cu mult mai numeroase decât încărcarea și vizualizarea imaginilor în XV, dar descrierea acestor facilități necesită sute de pagini. Dacă doriți să încercați celelalte caracteristici ale acestui program, vă rog să tipăriți documentația PostScript (așa cum este descris în această secțiune) și să o folosiți pentru a învăța mai multe despre XV. Altă modalitate, la fel de operativă, este să încercați pur și simplu acțiunea fiecărui buton și a fiecărei selecții de meniu din fereastra de control XV.

Figura A-11

Încarcarea
fișierului imagine
în fereastra
imagine a
programului XV.



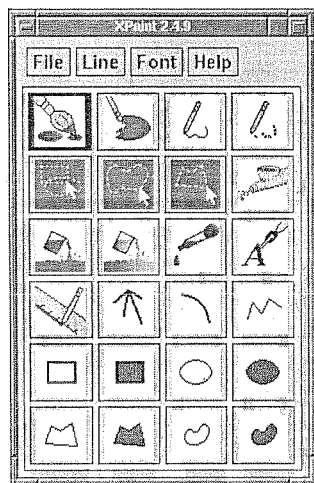
XPaint

XPaint este un program pentru afișarea și editarea imaginilor, ce a fost realizat având drept model venerabilul program MacPaint pentru Apple Macintosh. El rulează sub X Window System și vă permite să editați imagini sub formă de hartă de biți în câteva formate, inclusiv GIF, X11 Pixmap (xpm), X11 Bitmap (xbm) și TIFF.

XPaint este inclus pe CD-ROM-ul anexat acestei cărți. Puteți opta pentru instalarea acestui program atunci când instalați Linux de pe CD-ROM; se află în pachetul Graphics Manipulation. Pentru a lansa în execuție programul XPaint, tastați **xpaint** & într-o fereastră xterm. De pe suprafața de lucru a ferestrei X, puteți selecta din meniul Start->Programs->Graphics->xpaint. XPaint rulează și afișează fereastra ce conține caseta cu instrumente, așa cum se vede în figura A-12.

Figura A-12

Caseta cu
instrumente
pentru editarea
imaginilor în
XPaint



Toate operațiile de încărcare și editare ale imaginilor pornesc din fereastra ce conține caseta cu instrumente. Dacă ați mai folosit vreun program pentru editarea imaginilor, nu veți avea probleme cu instrumentele din casetă.

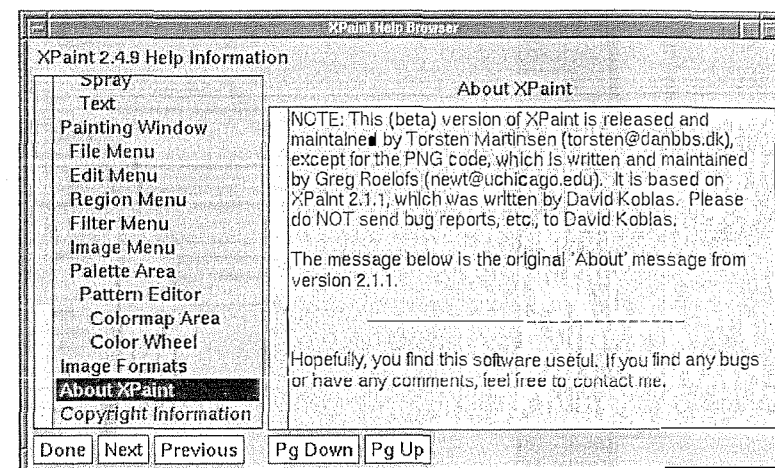
O bară de meniuri ce conține patru articole – File, Line, Font și Help – apare pe marginea superioară a casetei cu instrumente de lucru din programul XPaint. Dacă executați clic pe oricare dintre aceste articole, apare un meniu pop-up, din care puteți selecta în continuare. Meniul Help oferă câteva informații de asistență online.

Figura A-13 prezintă rezultatul selectării etichetei About din meniul Help.

Aceeași fereastră furnizează informații despre folosirea diferitelor caracteristici ale programului XPaint. Executați clic pe butoanele Next și Previous (din marginea inferioară a ferestrei de asistență) pentru a selecta un subiect și a examina informațiile aferente acestuia.

Figura A-13

Fereastra About
XPaint



xfig

xfig este un program de desenare interactiv ce rulează sub X și poate genera fișiere Encapsulated PostScript (EPS) potrivite pentru a fi incluse în documente. Pe CD-ROM-ul anexat acestei cărți, xfig se află în subdirectorul RedHat/RPMS în două fișiere Red Hat Package Manager (RPM): xfig-3.2-3.i386.rpm și transfig-3.2-4.i386.rpm.

Pentru a instala programul xfig, parcurgeți următorii pași:

1. Intrați în sistem ca root, introduceți CD-ROM-ul în unitate și tastați comanda:

```
mount /dev/cdrom /mnt/cdrom
```

2. Intrați în directorul adecvat cu următoarea comandă:

```
cd /mnt/cdrom/RedHat/RPMS
```

3. Instalați programul transfig utilizând comanda rpm, astfel:

```
rpm -ivh transfig*.rpm
```

4. Instalați programul xfig utilizând comanda rpm, astfel:

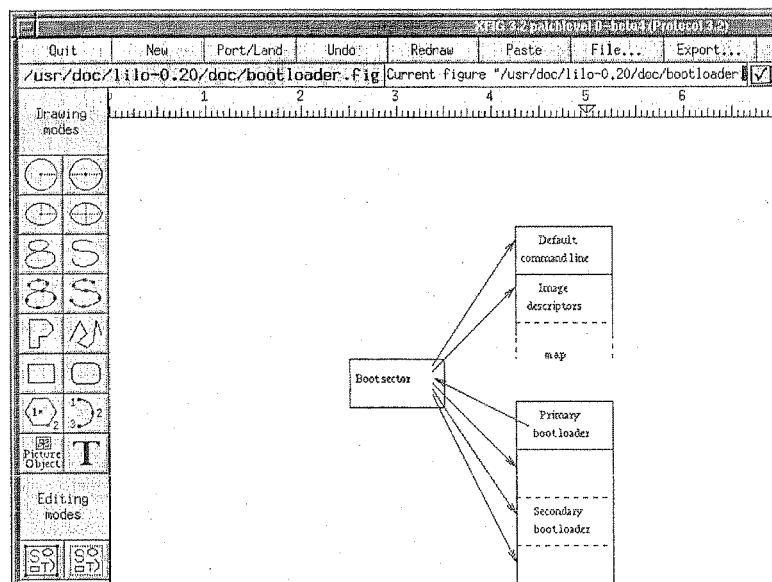
```
rpm -ivh xfig*.rpm
```

Pentru a încerca programul xfig, tastați **xfig** & într-o fereastră xterm. Această comandă determină apariția unei ferestre destul de mari. Pe marginea superioară a acestei ferestre veți găsi o bară de meniuri. Pentru a deschide un desen xfig, executați clic pe butonul File. Această acțiune aduce în prim-plan o casetă de dialog pentru selectarea fișierelor, prin intermediul căreia puteți să navigați printre directoare și să localizați fișierele xfig (care au de obicei extensia .fig).

Programul Linux Loader (LILO) folosește câteva desene xfig în documentația sa. Puteți găsi aceste fișiere în directorul /usr/doc/lilo-0.20/doc. Folosiți caseta de dialog pentru selectarea fișierelor din programul xfig pentru a localiza acest director și a selecta unul din fișiere. Figura A-14 prezintă fereastra xfig după încărcarea unui desen.

Figura A-14

Un desen demonstrativ afișat de programul xfig



Așa cum se vede în figura A-14, un grafician experimentat poate realiza desene impresionante cu acest program.

Ca în oricare altă aplicație, învățarea folosirii tuturor caracteristicilor xfig necesită exercițiu. Dacă sunteți familiarizat cu alte programe de desenare, precum MacDraw (Macintosh) sau CorelDRAW (PC), puteți folosi xfig fără prea mari probleme.

Gnuplot

Gnuplot este un utilitar de plotare interactiv. Trebuie să rulați Gnuplot sub X Window System, deoarece el folosește o fereastră X ca dispozitiv de ieșire. Gnuplot este un program condus prin linie de comandă; el vă solicită și vă acceptă comenzile pe care le introduceți. Ca răspuns la acele comenzi, Gnuplot afișează diferite tipuri de ploturi. Rezultatul apare într-o fereastră X. Întâmplător, chiar dacă Gnuplot are Gnu inclus în numele lui, el nu are nimic în comun cu GNU sau cu Free Software Foundation.

Pe CD-ROM-ul anexat acestei cărți, Gnuplot se află în fișierul `gnuplot-3.5-8.i386.rpm` din subdirectorul `RedHat/RPMS`.

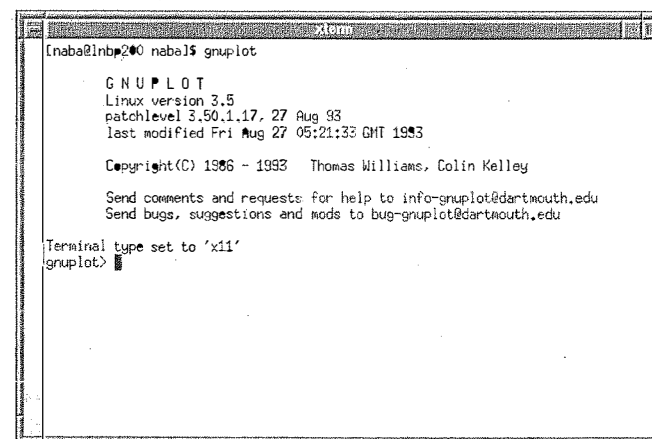
Pentru a instala Gnuplot, parcurgeți următorii pași:

1. Intrați în sistem ca `root`, introduceți CD-ROM-ul în unitate și tastați următoarea comandă:
`mount /dev/cdrom /mnt/cdrom`
2. Intrați în directorul adecvat cu următoarea comandă:
`cd /mnt/cdrom/RedHat/RPMS`
3. Instalați Gnuplot cu următoarea comandă `rpm`:
`rpm -ivh gnuplot*.rpm`

După instalare, ar trebui să lansați în execuție aplicația Gnuplot tastând `gnuplot` într-o fereastră `xterm`. Aplicația afișează un mesaj de deschidere și așteaptă să introduceți date, așa cum arată Figura A-15.

Figura A-15

Rularea aplicației Gnuplot într-o fereastră `xterm`



Pentru a vedea un rezultat imediat, tastați următoarea comandă Gnuplot la promptul afișat:

```
plot sin(x)
```

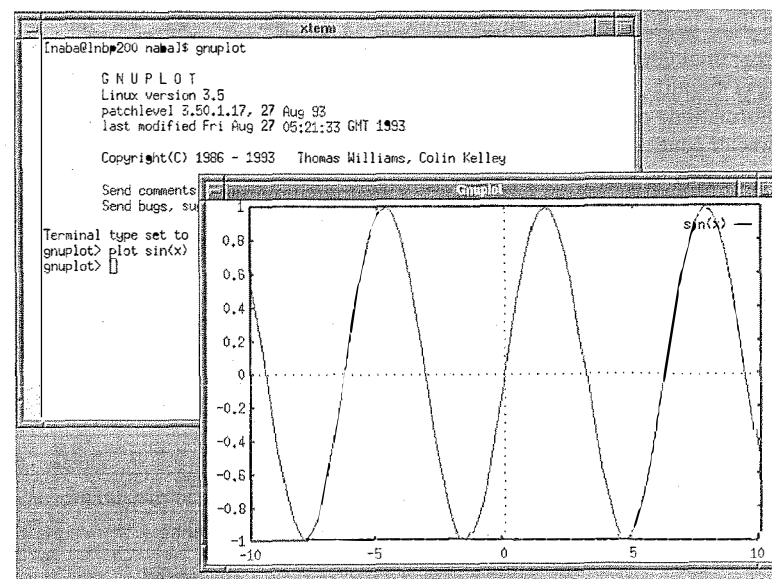
Gnuplot deschide o fereastră de ieșire și afișează un grafic al funcției sinus, ca în figura A-16.

Pentru a părăsi aplicația, executați clic pe fereastra `xterm` pentru a o activa și apoi tastați `quit`.

Acest exemplu este o simplă ilustrare a facilităților programului Gnuplot.

Figura A-16

Trasarea graficului funcției `sin(x)` în Gnuplot



În orice moment, în Gnuplot puteți apela asistența online care este asemănătoare cu cea a sistemului VAX/VMS din DEC. Pentru a învăța mai multe despre comanda `plot`, de exemplu, tastați `help plot` la promptul Gnuplot. Figura A-17 prezintă informația de asistență rezultantă.

Figura A-17

Asistență online
în Gnuplot

```

gnuplot> help plot
'plot' and 'splot' are the primary commands of the program. They plot
functions and data in many, many ways. 'plot' is used to plot 2-d
functions and data, while 'splot' plots 3-d surfaces and data.

Syntax:

plot <ranges> <<function> | <["<datafile">] {using ...}>>
    {title} {style} [, <function> {title} {style}...]

splot <ranges> <<function> | <["<datafile">] {index i} {using ...}>>
    {title} {style} [, <function> {title} {style}...]

where either a <function> or the name of a data file enclosed in quotes is
supplied. A function is a mathematical expression, or a pair ('plot') or
triple ('splot') of mathematical expressions in the case of parametric
functions. User-defined functions and variables may also be defined here.

'plot' and 'splot' commands can be as simple as

    plot sin(x)

and
Press return for more:
  
```

Ghostscript

Ghostscript este un utilitar pentru vizualizarea și tipărirea documentelor PostScript. El vă permite să tipăriți documente PostScript pe mai multe dispozitive non-PostScript.

Nucleul Ghostscript este o implementare aproape completă a limbajului PostScript. Ghostscript conține interpretorul care procesează intrarea PostScript și generează ieșirea pe un dispozitiv de ieșire. Un dispozitiv Ghostscript poate fi o imprimantă (sau un monitor), precum și un format de fișier imagine, precum BMP sau PCX.

Programul Ghostscript este distribuit sub GNU General Public License, dar este protejat de legea dreptului de autor și este întreținut de Aladdin Enterprises. Puteți găsi cele mai recente informații de contact în fișierul `README` din directorul `/usr/doc/ghostscript-3.33`. Acest director conține, de asemenea, documentație Ghostscript.

Documentația din directorul `/usr/doc/ghostscript-3.33` conține fișierele prezentate în tabelul A-1. Toate aceste fișiere sunt de tip text și pot fi vizualizate folosind comanda `more`.

Tabelul A-1 Documentația Ghostscript din directorul `/usr/doc/ghostscript-3.33`

Fișier	Descriere
<code>devices.doc</code>	Descrie detaliat câteva imprimante pe care Ghostscript poate tipări.
<code>drivers.doc</code>	Descrie interfața dintre Ghostscript și driverele de dispozitiv.
<code>fonts.doc</code>	Descrie fonturile și facilitățile acestora, furnizate o dată cu Ghostscript.
<code>hershey.doc</code>	Oferă informații despre fonturile Hershey.
<code>history1.doc</code>	Furnizează istoria modificărilor versiunilor Ghostscript mai vechi.
<code>history2.doc</code>	Furnizează modificările recente din Ghostscript.
<code>humor.doc</code>	Furnizează mesaje umoristice despre Ghostscript.

Fișier	Descriere
<code>language.doc</code>	Descrie limbajul Ghostscript.
<code>lib.doc</code>	Descrie biblioteca Ghostscript – o colecție de funcții C care implementează facilitățile pentru primitive grafice ale limbajului Ghostscript.
<code>make.doc</code>	Descrie modul de instalare a Ghostscript și de construire a executabilelor Ghostscript din codul sursă.
<code>NEWS</code>	Furnizează rezumatul modificărilor recente din Ghostscript.
<code>ps2epsi.doc</code>	Descrie utilitarul <code>ps2epsi</code> , care convertește un fișier Ghostscript la formatul Encapsulated Postscript Interchange din Adobe sau EPSI. (Puteți insera fișiere EPSI în multe procesoare de text.)
<code>psfiles.doc</code>	Descrie fișierele <code>.ps</code> din Ghostscript. (Aceste fișiere se află în directorul <code>/usr/share/ghostscript/3.33</code> .)
<code>use.doc</code>	Descrie modul de folosire al interpretorului limbajului Ghostscript.
<code>xfonts.doc</code>	Descrie interfața dintre Ghostscript și rutinele care accesează fonturile livrate extern și facilitățile de afișare ale textului.

Această documentație este elocventă în ceea ce privește facilitățile programului Ghostscript și ceea ce se livrează o dată cu acest program.

Pentru a rula Ghostscript, tastați `gs` într-o fereastră `xterm`. Aplicația aduce în prim-plan o fereastră vidă și afișează următorul text într-o fereastră `xterm`:

```

Aladdin Ghostscript 3.33 ( 4/10/1995 )
Copyright (C) 1995 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
This software comes with NO WARRANTY: see the file COPYING for details.
GS>
  
```

În acest punct, interactionați cu interpretorul Ghostscript. Dacă nu cunoașteți limbajul Ghostscript (care este ca PostScript), veți fi derutat la vederea acestui prompt. Este la fel ca promptul `c>` din DOS sau ca promptul de interpretor din UNIX afișat pe un terminal.

Dacă aveți un fișier PostScript disponibil, îl puteți încărca și vizualiza printr-o comandă simplă. De exemplu, încercați următoarea comandă:

```
GS> (/usr/share/ghostscript/3.33/examples/golfer.ps) run
```

Ceea ce ați tastat este o comandă Ghostscript care ar trebui să determine programul Ghostscript, să încarce fișierul `/usr/share/ghostscript/3.33/examples/golfer.ps` și să-l proceseze. Rezultatul este o imagine în fereastra de ieșire a programului Ghostscript. După aceasta, apăsați `Enter` și tastați `quit` pentru a părăsi Ghostscript.

Din fericire, nu este necesar să folosiți Ghostscript la nivel de interpretor (în afară de cazul în care cunoașteți bine PostScript și vreți să încercați interactiv comenzile PostScript). De obicei, folosiți GhostScript pentru a încărca și vizualiza un fișier PostScript.

Ghostscript preia câteva argumente în linia de comandă, inclusiv fișierul ce va fi încărcat. Pentru a vedea o listă a opțiunilor Ghostscript, tastați următoarea comandă:

```

gs -h
Copyright (C) 1995 Aladdin Enterprises, Menlo Park, CA. All rights reserved .
Usage: gs [switches] [file1 . ps file2 . ps . . . ]
Available devices:
    x11 x11alpha x11cmym x11mono deskjet djet500 djet500c
    dnj650c
  
```

```

laserjet ljetplus ljet2p ljet3 ljet3d ljet4 cdeskjet cdjcolor
cdjmono cdj500 cdj550 pj pjl pjl300 bj10e bj200
bjc600 epson eps9mid eps9high epsonc ibmpro jetp3852
dfaxhigh
dfaxlow faxg3 faxg32d faxg4 cp50 tiffg3 tiffg32d tiffg4
pcxmono pcxgray pcx16 pcx256 pcx24b pbm pbmraw pgm
pgmraw ppm pp-raw bit bitrgb bitcmyk tiffcrle tiffg3
tiffg32d tiffg4 tiff1zw tiffpack
Language interpreters:
PostScript PostScriptLevel1 PostScriptLevel2 PDF
Search path:
/usr/share/ghostscript/3.33/usr/share/ghostscript/fonts
Most frequently used switches : (you can use # in place of =)
-c quit (as the last switch) exit after processing
files
-d<name>[=<token>] define name as token, or true if no token
given
-DNOPAUSE don't pause between pages
-g<width>x<height> set width and height ('geometry'), in pixels
-Q 'quiet' mode, suppress most messages
-R<res> set resolution, in pixels per inch
-S<name>=<string> define name as string
-SDEVICE=<devname> select initial device
-sOutputFile=<file> select output file : embed %d or %ld for page
# ,
- means stdout, use | command to pipe
read from stdin ( e . g . , a pipe ) non-
interactively
For more information, see the ( plain text ) file use.doc in the
directory
/Usr/doc/ghostscript-3.33.

```

Figura A-18

Afișarea fișierului
/usr/share/ghostscript/3.33/
examples/
golfer.ps
de către
Ghostscript



Pentru a vedea cum Ghostscript randează un document PostScript, puteți folosi orice document PostScript disponibil. O soluție este să folosiți unul din fișierele PostScript de exemplificare din directorul /usr/share/ghostscript/3.33/examples. Tastați, de exemplu, următoarea comandă într-o fereastrăxterm:

```
gs
/usr/share/ghostscript/3.33/examples/golfer.ps
```

Ghostscript deschide acel fișier, îi procesează conținutul și afișează rezultatul în altă fereastră, ca în figura A-18.

În acest caz, rezultatul este imaginea unui jucător de golf.

După afișarea rezultatului, Ghostscript afișează următorul mesaj:

```
>showpage, press <return> to continue<<
```

Apăsați Enter pentru a continua. Pentru un document PostScript cu mai multe pagini, Ghostscript afișează pagina următoare. După ce sunt afișate toate paginile, reveniți la promptul Ghostscript. Tastați **quit** pentru a ieși din Ghostscript.

Ghostview

Ghostview este un front end grafic bazat pe X al interpretorului Ghostscript. Acest program este ideal pentru vizualizarea și tipărirea documentelor PostScript. Pentru un document lung, puteți chiar să tipăriți anumite pagini selectate. De asemenea, puteți vizualiza documentul la câteva niveluri de mărime (puteți mări sau micșora scara de vizualizare).

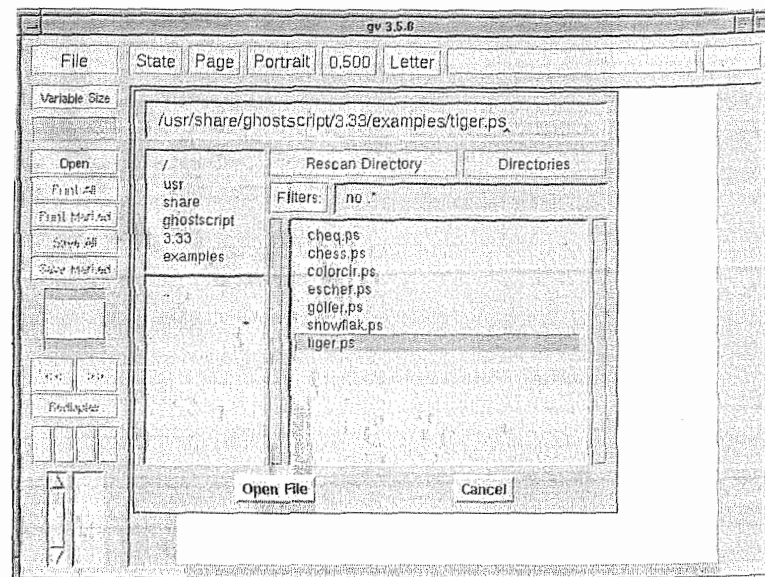
Pentru a lansa în execuție programul Ghostview, tastați **ghostview & în** o fereastră xterm. Această comandă determină apariția ferestrei Ghostview. Fereastra este împărțită în trei:

- De-a lungul marginii superioare veți găsi opt butoane. Primele cinci sunt butoane de meniu – când executați clic pe oricare din aceste butoane, apare un meniu.
- Pe partea stângă sunt ceva mai multe butoane, casete de text pentru afișarea informației și bare pentru derularea imaginii.
- Suprafața mare care ocupă cea mai mare parte a ferestrei Ghostview este suprafața de lucru, în care Ghostview afișează documentul PostScript.

Pentru a încărca și vizualiza un document PostScript în Ghostview, executați clic pe butonul File și selectați Open din meniul pop-up. Această acțiune determină Ghostview să afișeze o casetă de dialog pentru selectarea unui fișier. Folosiți această casetă de dialog pentru a selecta fișierul tiger.ps din directorul /usr/share/ghostscript/3.33/examples, așa cum se vede în figura A-19.

Figura A-19

Selectarea unui
fișier pentru a fi
deschis în
Ghostview



În figura A-19 vedeți în prim-plan caseta de dialog pentru selectarea fișierelor; fereastra Ghostview apare în fundal.

Pentru a deschide fișierul selectat, executați clic pe butonul Open File din caseta de dialog prezentată în figura A-19. Ghostview deschide fișierul selectat, îi procesează conținutul și afișează rezultatul în fereastra acestuia, așa cum se vede în figura A-20.

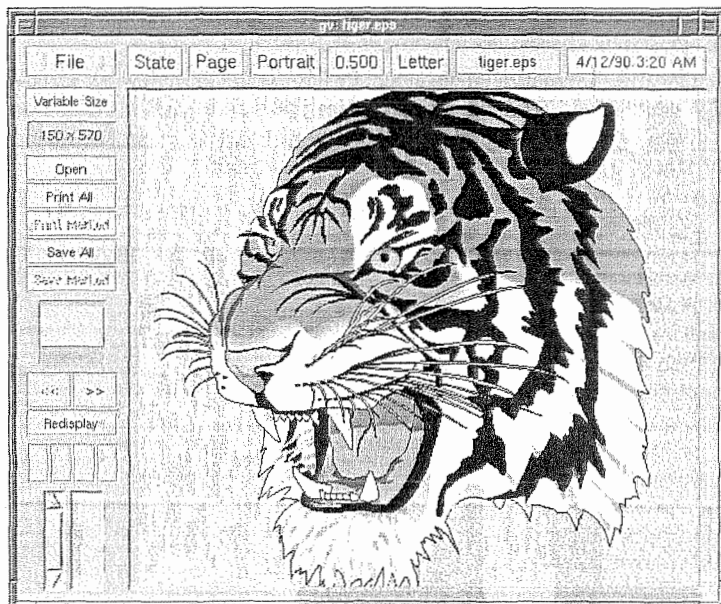
În ultimele două butoane din colțul dreapta-sus al ferestrei Ghostview afișează numele fișierului curent și data când acesta a fost creat. Ghostview preia aceste informații din comentariile aflate în fișierul PostScript, nu din marca de timp a fișierului.

Pe măsură ce deplasați mouse-ul deasupra imaginii, Ghostview afișează coordonatele indicatorului de mouse într-un buton situat de-a lungul colțului din stânga-sus al ferestrei.

Ghostview este folositor pentru vizualizarea diferitelor documentații în format PostScript (aceste fișiere au de obicei extensia .ps). De exemplu, am folosit Ghostview pentru a vizualiza documentația pentru programul XV; această documentație vine într-un fișier PostScript (/usr/doc/xv-3.10a/xvdocs.ps). Când vizualizez astfel de documente în Ghostview, folosesc butonul de mărire – butonul cu eticheta 0.500 din figura A-20. Dacă executați clic pe acest buton apare un meniu cu câțiva factori de mărire de la 0,1 la 10,0. Ar trebui să selectez un factor de mărire de 2.000 pentru a face documentul vizibil pe ecran.

Figura A-20

Afișarea fișierului /usr/share/ghostscript/3.33/examples/tiger.ps cu aplicația Ghostview.



Anexa B

Comenzi Linux

Această anexă prezintă o referință în ordine alfabetică a celor mai importante comenzi din Linux. Scopul este de a vă oferi o imagine de ansamblu asupra tuturor comenzilor necesare administrării fișierelor și directoarelor, pornirii și opririi proceselor, găsirii fișierelor, lucrului cu fișierele text și accesării asistenței online.

În cazul în care căutați o comandă pentru o anumită sarcină și nu știți care este această comandă, puteți considera folositoare parcurgerea comenzilor pe categorii.

Tabelul B-1 Comenzi Linux organizate pe categorii

Nume comandă	Acțiune
Apelarea asistenței online	
apropos	Găsește paginile man pentru un cuvânt cheie specificat
info	Afișează mesajul de asistență online despre o anumită comandă
man	Afișează mesajul de asistență online
whatis	Asemănătoare comenzii apropos, dar caută numai cuvinte complete
Utilizarea mai simplă a comenzilor	
alias	Definește un pseudonim pentru o comandă lungă
type	Afișează tipul și locația unei comenzi
unalias	Șterge un pseudonim definit prin comanda alias
Administrarea fișierelor și directoarelor	
cd	Schimbă directorul curent
chmod	Schimbă drepturile de acces la fișier
chown	Schimbă proprietarul și grupul fișierului
cp	Copiază fișiere
ln	Creează legături simbolice cu fișierele și directoarele
ls	Afișează conținutul unui director
mkdir	Creează un director
mv	Redenumeste sau mută un fișier dintr-un director în altul
rm	Șterge fișiere
rmdir	Șterge directoare
pwd	Afișează directorul curent
touch	Actualizează marca de timp a unui fișier
Găsirea fișierelor	
find	Găsește fișiere în funcție de anumite criterii cum ar fi nume, mărime etc.

(continuare)

Tabelul B-1 Continuare

Nume comanda	Acțiune
Găsirea fișierelor	
locate	Găsește fișiere utilizând o bază de date actualizată periodic
whereis	Găsește fișiere aflate în directoarele tipice unde sunt localizate fișierele executabile (cunoscute, de asemenea, ca fișiere binare)
which	Găsește fișiere în directoarele listate în variabila de mediu PATH
Procesarea fișierelor	
cat	Afișează un fișier la ieșirea standard (poate fi folosită la concatenarea mai multor fișiere într-un singur fișier mare)
cut	Extrage secțiunile specificate din fiecare linie de text a unui fișier
dd	Copiază blocuri de date dintr-un fișier în altul (folosită la copierea datelor din dispozitive)
diff	Compară două fișiere text și găsește toate deosebirile dintre ele
expand	Convertește toți tabulatorii în spații
file	Afișează tipul de date dintr-un fișier
fold	Formatează fiecare linie de text la o anumită lățime
grep	Caută expresii regulate într-un fișier text
less	Afișează un fișier text pagină cu pagină (se poate parcurge fișierul și în sens invers)
lpr	Tipărește fișiere
more	Afișează un fișier text pagină cu pagină (numai în nalt)
n1	Numără toate liniile dintr-un fișier text care conțin cel puțin un caracter și le tipărește la ieșirea standard
paste	Concatenează liniile corespondente din mai multe fișiere
patch	Actualizează un fișier text folosind deosebirile dintre original și copia revizuită a fișierului
sed	Copiază un fișier la ieșirea standard în timpul aplicării unor anumite comenzi de editare
sort	Sortează liniile într-un fișier text
split	Împarte un fișier în câteva fișiere mai mici, de o anumită mărime
tac	Inversează ordinea liniilor într-un fișier (ultima linie trece în locul primei linii etc.)
tail	Afișează ultimele linii dintr-un fișier
tr	Înlocuiește un grup de caractere cu altul într-un fișier întreg
uniq	Elimină liniile duplicate dintr-un fișier text
wc	Contorizează numărul de linii, cuvinte și caractere dintr-un fișier text
zcat	Afișează un fișier comprimat (după decompresare)
zless	Afișează un fișier comprimat pagină cu pagină (se poate parcurge fișierul și în sens invers)
zmore	Afișează un fișier comprimat pagină cu pagină

Arhivarea și comprimarea fișierelor

compress	Comprimă fișiere
cpio	Copiază fișiere în și dintr-o arhivă
gunzip	Decomprimă fișierele comprimate cu GNU ZIP (gzip) sau comanda compress
gzip	Comprimă fișierele (este mai puternică decât comanda compress)
tar	Creează o arhivă de fișiere conținute în unul sau mai multe directoare (inițial destinată pentru arhivarea pe bandă)
uncompress	Decomprimă fișiere comprimate prin comanda compress

Administrarea proceselor

bg	Rulează în fundal un proces întrerupt (background)
fg	Rulează un proces în prim-plan (foreground)
free	Afișează cantitatea de memorie liberă și utilizată din sistem
halt	Încheie execuția Linux și oprește calculatorul
kill	Trimite un semnal la un proces (de obicei folosit la terminarea unui proces)
ldd	Afișează bibliotecile partajate, necesare pentru a rula un program
nice	Rulează un proces cu prioritate mai mică (denumită „nice mode”)
ps	Afișează lista proceselor care rulează în momentul respectiv
printenv	Afișează variabilele de mediu curente
pstree	Asemănătoare cu ps, dar arată clar relațiile părinte – copil
reboot	Oprește Linux și apoi repornește calculatorul
shutdown	Încheie execuția sistemului Linux
top	Afișează lista majorității proceselor care folosesc intensiv procesorul și memoria
uname	Afișează informații despre sistem și despre nucleul Linux

Administrarea utilizatorilor

chsh	Schimbă interpretorul de comenzi
groups	Tipărește lista grupurilor ce conțin un utilizator specificat
id	Afișează utilizatorul și identificatorul grupului pentru un nume de utilizator specificat
passwd	Schimbă parola
su	Realizează login sub numele altui utilizator sau ca root (când comanda este invocată fără argumente)

Administrarea fișierului sistem

df	Calculează spațiul liber din toate dispozitivele de stocare instalate
du	Afișează informații despre gradul de ocupare a discului
fdformat	Formatează o dischetă
fdisk	Partiționează un hard-disc

(continuare)

I tabelul B-1 Continuare

Nume comandă Acțiune

Administrarea fișierului sistem

fsck	Verifică și repară un sistem de fișiere
mkfs	Creează un nou sistem de fișiere
mknod	Creează un fișier de periferic
mkswap	Creează spațiu de schimb pentru Linux într-un fișier sau într-o partiție de disc.
mount	Instalează un dispozitiv (de exemplu CD-ROM-ul) pe un director, în fișierul sistem
swapoff	Dezactivează un spațiu de schimb
swapon	Activează un spațiu de schimb
sync	Scrie datele din buffere în fișiere
tty	Afișează numele dispozitivului pentru terminalul curent
umount	Dezinstalează un dispozitiv din fișierul sistem

Lucrul cu data și ora

cal	Afișează calendarul pentru o anumită lună sau an
date	Arată data și ora curentă sau setează o nouă dată și oră

Restul acestei anexe prezintă intrări ale referințelor individuale pentru fiecare comandă prezentată în tabelul B-1. Fiecare intrare a unei referințe are un format standard ce conține următoarele secțiuni:

- **Scop:** Vă precizează când să utilizați comanda.
- **Sintaxă:** Prezintă sintaxa comenzii, cu câteva opțiuni uzuale. De asemenea, sunt prezentate și valorile opțiunilor tipice. Toate elementele opționale sunt incluse între paranteze pătrate.
- **Opțiuni:** Listează majoritatea opțiunilor, împreună cu o scurtă descriere pentru fiecare. Pentru multe comenzi veți găsi toate opțiunile listate în această secțiune. Totuși, unele comenzi au prea multe opțiuni de listat. Pentru acele comenzi vom prezenta cele mai utilizate opțiuni.
- **Descriere:** Descrie comanda și oferă mai multe detalii privind utilizarea acesteia.

alias

Scop	Definește un pseudonim pentru o comandă lungă sau vizualizează lista curentă de pseudonime
Sintaxă	alias [abbrev = command]
Opțiuni	Nici una
Descriere	Dacă tastați numai comanda alias, obțineți o listă cu toate <i>abrevierile</i> folosite curent. De obicei, puteți folosi comanda alias pentru a defini <i>abrevieri</i> ușor de reținut ale comenzilor mai lungi. De exemplu, dacă tastați deseori comanda <code>ls -l</code> puteți adăuga o linie ce conține <code>alias ll='ls -l'</code> în fișierul <code>.bashrc</code> din directorul dumneavoastră de bază. Apoi puteți tasta <code>ll</code> în loc de <code>ls -l</code> pentru a vizualiza o listă detaliată a directorului. alias este o comandă internă a interpretorului de comenzi Bash și este descrisă în capitolul 6.

apropos

Scop	Vizualizează lista tuturor paginilor man care conțin un anumit cuvânt cheie
Sintaxă	apropos cuvânt_cheie
Opțiuni	Nici una
Descriere	Comanda apropos caută cuvântul cheie într-o bază de date (cunoscută ca bază de date <i>whatis</i>) creată de programul <code>/usr/sbin/makewhatis</code> . Baza de date <i>whatis</i> este un index de cuvinte cheie conținute în toate paginile man din sistem. Din nefericire, când încercați comanda apropos cu un cuvânt cheie simplu precum <code>find</code> , puteți obține o lungă listă de pagini man, deoarece cuvântul <code>find</code> apare în multe asemenea pagini.

bg

Scop	Rulează un proces întrerupt în fundal
Sintaxă	bg
Opțiuni	Nici una
Descriere	După ce tastați o comandă ce necesită un timp îndelungat pentru a-și termina execuția, puteți tasta <code>CTRL+Z</code> pentru a întrerupe procesul. Apoi puteți tasta <code>bg</code> pentru a continua comanda în fundal în timp ce tastați alte comenzi la promptul interpretorului. <code>bg</code> este o comandă internă a interpretorului de comenzi Bash.

cal

Scop	Vizualizează calendarul oricărei luni, din orice an
Sintaxă	cal cal [-jy] [month_number] year]
Opțiuni	-j afișează datele conform calendarului iulian (cu numărul zilelor cuprins între 1 și 366) -y afișează calendarul pentru toate lunile anului curent

Descriere Dacă tastați `cal` fără alte opțiuni, comanda tipărește calendarul lunii curente. Dacă tastați `cal` urmat de un număr, comanda consideră numărul ca fiind anul și afișează calendarul pentru acel an. Pentru a vizualiza calendarul pentru o anumită lună dintr-un an, introduceți numărul lunii (1 = ianuarie, 2 = februarie etc.) urmat de numărul ce indică anul. Astfel, pentru a putea vizualiza calendarul lunii ianuarie a anului 2000, tastați următoarele:

cal 1 2000

```
January 2000
Su Mo Tu We Th Fr Sa
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

cat

Scop	Copiază conținutul unui fișier la ieșirea standard (ecranul)
Sintaxă	cat [-benstvA] files

cat

Opțiuni	-b numără liniile care conțin cel puțin un caracter -e arată sfârșitul unei linii (marcat prin caracterul \$) și toate caracterele netipăribile -n numără toate liniile rezultante începând cu numărul 1 -s înlocuiește mai multe linii vide cu o singură linie vidă -t arată tabulatoarele marcate prin caracterele ^I -v arată caracterele netipăribile -A arată toate caracterele (inclusiv pe cele netipăribile)
Descriere	De obicei, comanda <code>cat</code> este utilizată pentru a afișa conținutul unui fișier sau pentru a concatena numeroase fișiere într-unul singur. De exemplu, <code>cat file1 file2 file3</code> > <code>all</code> unește trei fișiere într-unul singur denumit <code>all</code> .

cd

Scop	Schimbă directorul curent
Sintaxă	<code>cd [directory]</code>
Opțiuni	Nici una
Descriere	Tastând <code>cd</code> fără numele directorului, directorul de bază devine director curent. Altfel, directorul specificat devine director curent. <code>cd</code> este o comandă internă a interpretorului de comenzi Bash.

chmod

Scop	Schimbă setările pentru accesul la unul sau mai multe fișiere															
Sintaxă	chmod [-cfvR] permission files															
Opțiuni	-c listează numai fișierele a căror permisiune de acces se schimbă -f oprește afișarea oricăror mesaje de eroare -v afișează în detaliu setările pentru accesul la fișiere -R schimbă recursiv setările pentru accesul la fișierele din toate directoarele															
Descriere	Pentru a folosi operativ comanda chmod, trebuie să învățați cum să specificați setările pentru acces. O modalitate este să concatenați câte o literă din fiecare din tabelele următoare în ordinea arătată (who, action, permission): <table><tr><th>Who</th><th>Action</th><th>Permission</th></tr><tr><td>u</td><td>user + add</td><td>r read</td></tr><tr><td>g</td><td>group - remove</td><td>w write</td></tr><tr><td>o</td><td>others = assign</td><td>x execute</td></tr><tr><td>a</td><td>all</td><td>s set user ID</td></tr></table> Pentru a permite tuturor accesul la citire pentru toate fișierele dintr-un director, ar trebui să tastezi chmod a+ r *. Pe de altă parte, pentru a face ca un anumit fișier să poată fi apelat de către oricine, ar trebui să tastezi chmod +x filename. Altă modalitate pentru a preciza setările drepturilor de acces este să utilizați o succesiune de trei cifre a numerelor în baza opt. Într-o listă detaliată, accesul la citire, scriere și execuție pentru utilizator (user), grup (group) și alții (others) apare ca o succesiune rwxrwxrwx (cu liniuțe în loc de litere pentru operațiile nepermise). Considerați rwxrwxrwx ca reprezentând de trei ori șirul rwx. Acum atribuim valorile r=4, w=2 și x=1. Pentru a obține valoarea secvenței rwx, adunăm valorile lui r, w și x. Astfel rwx = 7. Utilizând această formulă puteți atribui o valoare cu trei cifre oricărei setări de acces. De exemplu, dacă utilizatorul poate citi și scrie într-un fișier, iar ceilalți pot numai citi fișierul, setarea de acces este rw-r-r- (astfel apare în listă) și valoarea este 644. Astfel, dacă doriți ca toate fișierele dintr-un director să poată fi citite de oricine, dar să poată scrie în ele doar utilizatorul, ar trebui să utilizați comanda chmod 644*.	Who	Action	Permission	u	user + add	r read	g	group - remove	w write	o	others = assign	x execute	a	all	s set user ID
Who	Action	Permission														
u	user + add	r read														
g	group - remove	w write														
o	others = assign	x execute														
a	all	s set user ID														

chown

Scop	Schimbă utilizatorul și grupul care este proprietar al unui fișier
Sintaxă	<code>chown [cvfR] username.groupname files</code>
Opțiuni	-c listează numai fișierele care au proprietarul schimbat -f oprește afișarea oricăror mesaje de eroare -v afișează în detaliu schimbările de proprietate -R schimbă recursiv proprietarul fișierelor din toate subdirectoarele
Descriere	Pentru a face un utilizator proprietar al unui sau mai multor fișiere, invocați <code>chown</code> cu numele utilizatorului urmat de numele fișierului. La fel de bine, pentru a schimba proprietarul grupului adăugați numele noului grup la numele utilizatorului, cu un punct ca separator. De exemplu, pentru a face utilizatorul <code>naba</code> proprietar al tuturor fișierelor dintr-un director, voi tastea <code>chown naba*</code> . Observați că trebuie să intrați în sistem ca <code>root</code> pentru a schimba proprietarul fișierelor.

chsh

Scop	Schimbă interpretorul prestabilit lansat la intrarea în sistem
Sintaxă	<code>chsh [-s shell] [username]</code>
Opțiuni	-s specifică numele executabilului ce va fi folosit ca interpretor (poate fi orice program listat în fișierul <code>/etc/shells</code> , cum ar fi <code>/bin/bash/</code> și <code>/bin/csh</code>).
Descriere	Programul shell prestabilit al utilizatorului este stocat în fișierul <code>/etc/passwd</code>. Comanda <code>chsh</code> vă permite să schimbați programul shell prestabilit cu oricare din cele listate în fișierul <code>/etc/shells</code>. Dacă tastezi <code>chsh</code> fără alte opțiuni, vi se cere să introduceți numele programului shell.

compress

Scop	Comprimă unul sau mai multe fișiere, utilizând metoda de comprimare Lempel – Ziv
Sintaxă	<code>compress [-cdrvV] files</code>
Opțiuni	-c scrie fișierul comprimat la ieșirea standard și păstrează originalul -d decompimă fișierul -r comprimă recursiv fișierele din toate subdirectoarele -v afișează un mesaj pe măsură ce este comprimat fiecare fișier -V afișează numărul versiunii și iese din program
Descriere	Comanda <code>compress</code> comprimă fiecare fișier specificat și înlocuiește originalul cu versiunea comprimată (cu sufixul <code>.z</code> adăugat la nume). Puteți decompima fișierul utilizând comanda <code>compress -d</code> sau comanda <code>uncompress</code> .

cp

Scop	Copiază fișiere și directoare
Sintaxă	<code>cp [options] source_files destination_directory</code> <code>cp [options] sourcefile destination_file</code>

cp

Opțiuni	-a păstrează toate atributele fișierului -b face copii de siguranță înainte de copiere -l copiază o legătură, dar nu și fișierul la care face trimitere legătura -i solicită confirmare înainte de a suprascrie fișiere -l creează legături hard (hard links) în loc să copieze fișierele -p păstrează proprietatea, drepturile de acces și marca de timp a fișierului -R copiază recursiv fișierele din toate subdirectoarele -s creează legături soft în loc să copieze fișierele -u copiază numai atunci când fișierul care este copiat este mai nou decât cel peste care trebuie copiat -v afișează în detaliu mesaje pe măsură ce se realizează copierea -help afișează mesajul de asistență pentru comanda cp
----------------	---

Descriere	Comanda cp copiază un fișier în altul. De asemenea, puteți copia câteva fișiere dintr-un director în altul.
------------------	---

cpio

Scop	Copiază fișierele în și dintr-o arhivă care se poate găsi pe un mediu de stocare cum ar fi banda sau un fișier de pe disc
-------------	---

Sintaxă	<pre>cpio [-icdv] pattern cpio [-ocBv] pattern cpio [-padm] pattern</pre>
----------------	---

Opțiuni	-i extrage fișierele al căror nume se potrivesc cu modelul -o copiază în arhivă fișierele ale căror nume sunt furnizate la intrarea standard -p copiază fișierele în alt director, pe același sistem -a resetează timpii de acces ai fișierelor de intrare -B copiază utilizând 5120 de baiți pe înregistrare (în mod prestabilit se utilizează 512 baiți pe înregistrare) -c citește și scrie informațiile din antet sub formă de caractere ASCII -d creează directoarele necesare -m reține timpul de modificare al fișierului anterior -v afișează o listă cu nume de fișiere
----------------	--

Descriere	Comanda cpio copiază fișiere în și din arhive. Există trei variante distincte ale comenzii cpio: cpio -o creează o arhivă; cpio -i extrage dintr-o arhivă; cpio -p copiază dintr-un director în altul. cpio nu este atât de cunoscută de utilizatorii Linux. Comanda tar este mult mai utilizată. Totuși, unele programe de instalare utilizează cpio în timpul procesului de instalare.
------------------	--

cut

Scop	Copiază părți selectate din fiecare linie de text dintr-un fișier la ieșirea standard
-------------	---

Sintaxă	cut [options] file
----------------	----------------------

Opțiuni	-b list extrage caracterele aflate la pozițiile specificate în listă -f list extrage câmpurile (considerate a fi separate prin tabulatori) specificate în listă -d char specifică acel caracter care delimitează câmpurile (în mod prestabilit este tabulatorul) -s sare liniile care nu conțin câmpuri delimitate (vezi opțiunea -f)
----------------	--

cut

Descriere	Comanda cut specifică părți din fiecare linie de text dintr-un fișier și scrie acele linii la ieșirea standard. Mai puteți extrage un grup de caractere (specificate de pozițiile lor) din fiecare linie sau câmpuri, unde câmpurile sunt separate de un caracter special cum ar fi tabulatorul. De exemplu, pentru a extrage caracterele între 1 și 11 și al 56-lea caracter numărând înainte din lista detaliată a directoarelor, utilizați următoarea comandă:
------------------	---

```
1s -l | cut -b 1 - 11, 56 - | more
-rw-r--r-- -r- - DIR_COLORS
-rw-r--r-- -r- - HOSTNAME
-rw-r--r-- -r- - Muttrc
-rwxr-xr-x X11
-rw-r--r-- -r- - adjtime
-rw-r--r-- -r- - aliases
-rw-r--r-- -r- - aliases.db
-rw-r--r-- -r- - at.deny
( ... linii cterse)
```

date

Scop	Afișează data și ora curentă sau setează o dată și o oră nouă
-------------	---

Sintaxă	<pre>date [options] [+format] date [-su] [MMDDHHMM[CC]YY][.SS]]</pre>
----------------	---

Opțiuni	-s setează data și/sau ora -u afișează sau setează ora utilizând ora medie Greenwich (GMT)
----------------	---

Descriere	Comanda date fără opțiuni afișează data și ora curentă. Utilizând argumentul +format, puteți specifica formatul de afișare a datei și orei. Pentru o listă completă cu specificația formatului tastați man date.
------------------	--

Pentru a seta data, utilizați date -s urmată de data și ora în format MMDDHHMM, unde fiecare caracter reprezintă o cifră (MM este numărul lunii, DD este ziua, HH este ora și MM sunt minutele. La fel de bine puteți specifica, opțional, anul (YY) și secolul (CC).

dd

Scop	Copiază blocuri de date de la intrarea standard la ieșirea standard (și, opțional, convertește datele dintr-un format în altul)
-------------	---

Sintaxă	dd option1=value1 option2=value2 option3=value3 ...
----------------	---

Opțiuni	<pre>if=file citește din fișierul specificat în loc să citească de la intrarea standard of=file scrie în fișierul specificat în loc să scrie la ieșirea standard ibs=nbytes citește blocuri de câte n baiți obs=nbytes scrie blocuri de câte n baiți bs=nbytes citește și scrie blocuri de câte n baiți cbs=nbytes convertește blocuri de câte n baiți skip=nblocks sare n blocuri de intrare de la începutul fișierului de intrare seek=nblocks sare n blocuri de ieșire în fișierul de ieșire count=nblocks copiază n blocuri de la intrare la ieșire</pre>
----------------	---

(continuare)

conv=code efectuează conversia; code poate fi unul din următoarele:

ascii convertește EBCDIC în ASCII

ebcdic convertește ASCII în EBCDIC

lcase convertește în litere mici

ucase convertește în majuscule

swab schimbă între ei baiții din fiecare pereche de intrare

noerror continuă conversia după citirea erorilor

(Observație: EBCDIC este un format de codificare folosit în sistemele IBM mainframe.)

Descriere Comanda dd copiază blocuri de date de la intrarea standard la ieșirea standard, convertind opțional datele pe măsură ce se realizează copierea. De obicei, dd este utilizată pentru a copia datele direct de pe un dispozitiv pe altul. De exemplu, puteți copia nucleul Linux (/boot/vmlinuz) pe o dischetă, cu următoarea comandă:

```
dd if=/boot/vmlinuz of=/dev/fd0
```

df

Scop Afișează cantitatea de memorie liberă și utilizată din toate fișierele sistem instalate.

Sintaxă df [options] [filesystem]

Opțiuni

- a afișează informații despre toate fișierele sistem
- i afișează informațiile inode (discul este organizat în inodes)
- T tipărește tipul fișierului sistem
- t type afișează informații numai despre tipurile de fișiere sistem specificate
- x type nu afișează tipurile de fișiere sistem specificate
- help afișează mesajul de asistență

Descriere Comanda df afișează cantitatea de memorie liberă și utilizată dintr-un fișier sistem. Dacă doriți să știți cât de ocupate sunt discurile dumneavoastră, utilizați comanda df fără nici un argument. Apoi, comanda df afișează informații despre memoria utilizată și cea disponibilă din toate fișierele sistem instalate în momentul respectiv.

diff

Scop Prezintă deosebirea dintre două fișiere text (sau dintre toate fișierele cu același nume din două directoare)

Sintaxă diff [options] from_file to_file

Opțiuni

- a tratează toate fișierele ca fișiere text, chiar dacă nu par a fi fișiere text
- b ignoră liniile goale și spațiile libere repetate
- c afișează rezultatul într-un format diferit
- d încearcă să găsească un set mai mic de modificări (cu această opțiune comanda diff rulează mai lent)
- e creează un fișier script pentru editorul ed pentru a converti fișierul from_file în fișierul to_file
- f afișează rezultatul la fel ca la opțiunea -e, dar în ordine inversă
- i nu face distincție între literele mari și literele mici
- l transmite ieșirea către comanda pr pentru a fi paginată
- n lucrează la fel ca opțiunea -f, dar contorizează numărul liniilor modificate
- r compară recursiv fișierele cu același nume din toate subdirectoarele
- s raportează identitatea a două fișiere

- t detaliază tabulatoarele la spații, în rezultatul afișat
- u folosește formatul de afișare unificată
- v afișează versiunea comenzii diff
- w ignoră spațiile și tabulatoarele când compară liniile

Descriere Comanda diff compară fișierul from_file cu fișierul to_file și afișează liniile care diferă. Rezultatul poate fi într-un format pe care comanda patch îl poate utiliza pentru a converti fișierul from_file în to_file.

du

Scop Afișează informații sumare despre gradul de ocupare al discului, în kilobaiți

Sintaxă du [options] [directories_or_files]

Opțiuni

- a afișează informații despre mărimea tuturor fișierelor (nu numai a directoarelor)
- b afișează gradul de ocupare în baiți (în loc de kilobaiți)
- c afișează un total general al spațiului ocupat pe disc
- k afișează informații despre gradul de ocupare în kilobaiți (aceasta este opțiunea prestabilită)
- s afișează cantitatea de memorie liberă și ocupată pentru tot discul, fără detalierea pe directoare.

Descriere Comanda du afișează spațiul discului (în kilobaiți) utilizat de directoarele sau fișierele specificate. În mod prestabilit, această comandă afișează spațiul de pe disc utilizat de fiecare director și subdirector. O utilizare uzuală a comenzii du o reprezintă invocarea acesteia cu opțiunea -s pentru a vedea spațiul total utilizat de directorul curent. De exemplu, iată cum puteți verifica detaliile despre spațiul ocupat de directorul /usr/doc/HOWTO:

```
du -up /usr/doc/HOWTO
4742 /usr/doc/HOWTO/mini/other-formats/html
4742 /usr/doc/HOWTO/mini/other-formats/sgml
4742 /usr/doc/HOWTO/mini/other-formats/html
4742 /usr/doc/HOWTO/mini
4742 /usr/doc/HOWTO
```

expand

Scop Scrie fișiere la ieșirea standard, după detalierea fiecărui tabulator într-un număr corespunzător de spații

Sintaxă expand [options] [files]

Opțiuni

- n (unde n este un număr) setează tabulatoarele la un interval de n spații
- n1 [n2, ...] (unde n1, n2, ... sunt numere) specifică marcasele de tabulare
- i convertește numai tabulatorul inițial în spații

Descriere Comanda expand citește din fișierele specificate (sau din intrarea standard, dacă nu sunt specificate fișierele) și le scrie la ieșirea standard, cu fiecare tabulator detaliat la numărul corespunzător de spații. În mod prestabilit, expand consideră că pozițiile tabulatoarelor sunt la interval de opt spații (este echivalent cu opțiunea -8)

fdformat

Scop Formatează o dischetă specificată prin numele dispozitivului (cum ar fi /dev/fd0H1440 pentru o dischetă de 3,5 inch de densitate mare, din unitatea A)

Sintaxă fdformat [-n] device_name

fdformat

Opțiuni	-n dezactivează verificarea efectuată după formatare
Descriere	Comanda <code>fdformat</code> formatează o dischetă. Utilizați un nume de dispozitiv adecvat pentru a identifica litera corespunzătoare unității de dischetă (vezi capitolul 7 pentru convențiile de nume ale unităților de disc). După formatarea dischetei cu <code>fdformat</code> , puteți utiliza comanda <code>mkfs</code> pentru a instala un fișier sistem Linux sau <code>tar</code> pentru a stoca o arhivă.

fdisk

Scop	Partiționează un disc sau afișează informații despre partițiile existente
Sintaxă	<code>fdisk [options] [device_name]</code>
Opțiuni	-l afișează tabelele de partiție și își încheie execuția -s device afișează mărimea partiției specificate -v afișează numărul versiunii pentru comanda <code>fdisk</code>
Descriere	Comanda <code>fdisk</code> partiționează un hard disc specificat (vezi Capitolul 11 pentru convențiile de denumire ale unității de disc). De asemenea, puteți utiliza <code>fdisk</code> pentru a afișa informații despre partițiile existente. Nu ar trebui să rulați niciodată <code>fdisk</code> și să schimbați partițiile unui hard-disc în timp ce una sau mai multe dintre partițiile lui sunt instalate pe fișierul sistem Linux. În schimb, ar trebui să inițializați calculatorul de pe o dischetă de instalare (sau o dischetă de inițializare) și apoi să realizați partiționarea. Amintiți-vă că, de obicei, partiționarea distruge toate datele existente pe un hard-disc.

fg

Scop	Continuă în prim-plan (foreground) un proces întrerupt
Sintaxă	<code>fg</code>
Opțiuni	Nici una
Descriere	După ce întrerupeți un proces tastând <code>Ctrl+Z</code> , îl puteți continua în prim-plan (foreground), introducând comanda <code>fg</code> . Observați că <code>fg</code> este o comandă internă a interpretorului de comenzi Bash.

file

Scop	Afișează tipul de date dintr-un fișier, bazându-se pe regulile definite în fișierul <code>/usr/lib/magic</code> (cunoscut, de asemenea, ca „fișierul magic”)
Sintaxă	<code>file [options] files</code>
Opțiuni	-c afișează o formă analizată a unui fișier magic (sau fișierul prestabilit) și își încheie execuția -m file1 [:file2:...] specifică alte fișiere magic -v afișează numărul versiunii și iese din comandă -z vizualizează conținutul fișierelor comprimate
Descriere	Comanda <code>file</code> utilizează regulile specificate în fișierul <code>/usr/lib/magic</code> pentru a determina tipul de date din fișierele specificate. De exemplu, puteți utiliza comanda <code>file</code> pentru a verifica tipul fiecărui fișier din directorul <code>/usr/lib</code> , după cum urmează:

file

```
file *|more
Mort1.o:          ELF 32-bit LSB relocatable, Intel
80386, version1, not stripped
X11:              symbolic link to ../X11R6/lib/X11
autoconf:         directory
bison.hairy:      C program text
bison.simple:     C program text
cracklib_dict.hwm: ASCII text
cracklib_dict.pwd: ASCII text
cracklib_dict.pwi: ASCII text
(...linii sterse)
```

find

Scop	Afișează lista fișierelor care îndeplinesc un anumit set de criterii
Sintaxă	<code>find [path] [options]</code>
Opțiuni	-depth procesează mai întâi directorul curent, iar apoi subdirectoarele -maxdepth n limitează căutarea la n niveluri ale directorilor -follow procesează directoarele incluse prin legături simbolice -name pattern găsește fișiere ale căror nume se potrivesc cu modelul (pattern) -ctime n localizează fișierele modificate, exact cu n zile în urmă -user uname găsește fișierele aparținând unui utilizator specificat -group gname găsește fișierele aparținând de un grup specificat -path pattern găsește fișierele a căror cale se potrivește cu modelul -perm mode găsește fișierele cu setarea de acces specificată -size +nK găsește fișierele mai mari de n kilobați -type x găsește fișierele de un anumit tip, unde x poate fi: f pentru fișiere d pentru directoare l pentru legături simbolice -print afișează numele fișierelor găsite -exec command [options] {} \ ; execută comanda specificată, furnizându-i ca argument numele fișierului găsit.
Descriere	Comanda <code>find</code> este utilizată pentru găsirea tuturor fișierelor ce îndeplinesc un set de criterii. Dacă tastați <code>find</code> fără alte argumente, rezultatul este o listă cu toate fișierele din toate subdirectoarele directorului curent. Pentru a vedea toate fișierele al căror nume se termină cu <code>.gz</code> , ar trebui să tastați <code>find . -name *.gz</code> .

fold

Scop	Formatează toate liniile de text la o anumită lățime (lățimea prestabilită este de 80 de caractere)
Sintaxă	<code>fold [options] [files]</code>
Opțiuni	-b numără bați-li în locul coloanelor, deci caracterele backspace și tabulatoarele sunt numărate -s întrerupe liniile fără să împartă vreun cuvânt în silabe -w N (unde N este un număr) setează linia la o lățime de N caractere
Descriere	Comanda <code>fold</code> formatează fiecare linie introdusă la un număr specificat de caractere și afișează rezultatul pe ecran (ieșirea standard). Dacă nu specificați numele vreunui fișier, <code>fold</code> citește liniile de la intrarea standard.

free	
Scop	Afișează cantitatea de memorie liberă și utilizată din sistem
Sintaxă	<code>free [options]</code>
Opțiuni	-b afișează memoria în baiți -k afișează memoria în kilobaiți (opțiune prestabilită) -m afișează memoria în megabaiți -s <i>n</i> repetă comanda la fiecare <i>n</i> secunde -t afișează o linie ce conține repartitia tuturor blocurilor de memorie
Descriere	Comanda <code>free</code> afișează informații despre memoria fizică (RAM) și despre zona de memorie virtuală (de pe disc). Rezultatul arată cantitatea totală de memorie, atât cea liberă cât și cea utilizată.
fsck	
Scop	Verifică și repară un sistem de fișiere Linux
Sintaxă	<code>fsck [options] device_name</code>
Opțiuni	-A verifică toate sistemele de fișiere listate în fișierul <code>/etc/fstab</code> -R omite sistemul de fișiere <code>root</code> (când verifică toate sistemele de fișiere) -T nu arată titlul la pornire -N arată ce ar putea fi făcut, dar, de fapt, nu face nimic -V emite un rezultat detaliat -t <i>fstype</i> specifică tipul sistemului de fișiere (de exemplu <code>ext2</code>) -n răspunde negativ la toate solicitările de confirmare (numai pentru sistemul de fișiere <code>ext2</code>) -p realizează toate reparațiile fără să ceară confirmare (pentru <code>ext2</code>) -y răspunde pozitiv la toate solicitările de confirmare cu NU (numai pentru sistemul de fișiere <code>ext2</code>)
Descriere	Comanda <code>fsck</code> verifică integritatea unui sistem de fișiere și realizează orice reparație necesară. În funcție de tipul sistemului de fișiere, <code>fsck</code> rulează o comandă adecvată pentru a îndeplini sarcina de verificare și reparare a sistemului de fișiere. De exemplu, pentru a verifica un sistem de fișiere <code>ext2</code> , comanda <code>fsck</code> rulează programul <code>e2fsck</code> . Trebuie să lansați în execuție <code>fsck</code> când deconectați sistemul, fără să rulați comanda <code>shutdown</code> . De obicei, această comandă este lansată în execuție automat în timpul procedurii de pornire a sistemului.
grep	
Scop	Caută unul sau mai multe fișiere pentru a găsi linii care conțin o expresie regulată (un model de căutare)
Sintaxă	<code>grep [options] pattern files</code>
Opțiuni	-N (unde <i>N</i> este un număr) afișează <i>N</i> linii din jurul liniei ce conține modelul de căutare (<i>pattern</i>) -c afișează numărul de linii care conțin modelul de căutare -f <i>file</i> citește opțiunile din fișierul specificat -i nu face distincția între majuscule și literele mici -l afișează numele fișierelor care conțin modelul de căutare (<i>pattern</i>) -n afișează numărul liniilor pentru liniile care conțin modelul -q returnează un cod de stare, dar nu afișează nimic -v afișează liniile care nu conțin modelul de căutare -w găsește numai cuvintele întregi care se potrivesc modelului

Descriere	Comanda <code>grep</code> caută în fișierele specificate un model. Modelul (<i>pattern</i>) este o expresie regulată care își are propriile reguli. De obicei, utilizați <code>grep</code> pentru a căuta o anumită succesiune de caractere într-unul sau mai multe fișiere text.
groups	
Scop	Arată grupurile de care aparține un utilizator
Sintaxă	<code>groups [username]</code>
Opțiuni	Nici una.
Descriere	Comanda <code>groups</code> afișează numele grupurilor de care aparține un utilizator. Dacă nu specificați numele utilizatorului, comanda afișează grupurile din care faceți parte dumneavoastră
gunzip	
Scop	Decomprimă fișierele comprimate cu <code>gzip</code> sau <code>compress</code>
Sintaxă	<code>gunzip [options] files</code>
Opțiuni	Vezi opțiunile pentru <code>gzip</code>
Descriere	Comanda <code>gunzip</code> decompimă fișierele comprimate (aceste fișiere au extensia <code>.gz</code> sau <code>.Z</code>). După decomprimare <code>gunzip</code> înlocuiește fișierele comprimate cu versiunile lor decomprimare și elimină din numele fișierelor extensia <code>.gz</code> sau <code>.Z</code> . Comanda <code>gunzip</code> este identică cu <code>gzip</code> cu opțiunea <code>-d</code> .
gzip	
Scop	Comprimă unul sau mai multe fișiere
Sintaxă	<code>gzip [options] files</code>
Opțiuni	-c scrie rezultatul la ieșirea standard și reține fișierul original -d decompimă fișierul (la fel ca <code>gunzip</code>) -h afișează un mesaj de asistență -l listează conținutul unei fișier comprimat -n nu salvează numele original și marca de timp -r comprimă recursiv fișierele din toate subdirectoarele -v afișează rezultatul detaliat -V afișează numărul versiunii
Descriere	Comanda <code>gzip</code> comprimă fișierele utilizând codificarea Lempel-Ziv (LZ77) care realizează mai bine comprimarea decât algoritmul utilizat de comanda <code>compress</code> . După comprimarea unui fișier, <code>gzip</code> înlocuiește fișierul original cu versiunea comprimată și adaugă <code>.gz</code> la numele fișierului.
halt	
Scop	Termină toate procesele și oprește sistemul (trebuie să vă conectați ca <code>root</code> – rădăcină)
Sintaxă	<code>halt [options]</code>
Opțiuni	-n -f forțează oprirea fără apelarea comenzii <code>/sbin/shutdown</code> -i închide toate interfețele de rețea înainte de oprirea sistemului
Descriere	Comanda <code>halt</code> permite superutilizatorului (<code>root</code>) să încheie toate procesele și oprește sistemul. Comanda <code>halt</code> apelează comanda <code>/sbin/shutdown</code> cu opțiunea <code>-h</code> .

id

Scop	Listează identificatorul (ID) utilizatorului, identificatorul grupului și grupurile pentru un utilizator
Sintaxă	<code>id [options] [username]</code>
Opțiuni	-g afișează numai identificatorul grupului -n afișează numele grupului în locul identificatorului -u afișează numai identificatorul utilizatorului
Descriere	Comanda <code>id</code> afișează identificatorul utilizatorului, identificatorul grupului și grupurile pentru un utilizator specificat. Dacă nu precizați numele utilizatorului, <code>id</code> afișează informații despre utilizatorul curent.

info

Scop	Vizualizează informații de asistență online despre orice comandă Linux
Sintaxă	<code>info [options] command</code>
Opțiuni	-d <i>dirname</i> adaugă un director la lista de directoare în care se vor căuta fișierele -f <i>infofile</i> specifică fișierul ce va fi utilizat de comandă -h afișează informații de utilizare a comenzii
Descriere	Comanda <code>info</code> afișează informații de asistență online despre o comandă specificată, într-o fereastră text pe tot ecranul. Puteți utiliza comenzi Emacs pentru a naviga prin textul afișat de comanda <code>info</code> . Pentru a învăța mai multe despre această comandă, tasteți <code>info</code> fără nici un argument.

kill

Scop	Trimite un semnal la un proces
Sintaxă	<code>kill [options] process_id</code>
Opțiuni	-s <i>signal</i> (unde <i>signal</i> este un număr sau un nume) trimite semnalul specificat -l listează numele și numerele semnalului
Descriere	Comanda <code>kill</code> trimite un semnal la un proces. De obicei, semnalul are rolul de a opri procesul. De exemplu, <code>kill -9 123</code> termină procesul cu identificatorul (ID) 123. Pentru a vedea identificatoarele procesului utilizați comanda <code>ps</code> . Pentru a vedea o listă de nume și numere de semnale, tasteți <code>kill -l</code> (adică litera „l” mic)

ldd

Scop	Afișează numele bibliotecilor partajate, necesare pentru a rula un program
Sintaxă	<code>ldd [options] programs</code>
Opțiuni	-v afișează numărul versiunii comenzii <code>ldd</code> -V afișează numărul versiunii pentru programul ce realizează legătura dinamică (<code>ld.so</code>) -d relocalizează funcțiile și le raportează pe cele care lipsesc -r relocalizează atât datele, cât și funcțiile, și raportează obiectele care lipsesc
Descriere	Comanda <code>ldd</code> vă permite să determinați ce biblioteci partajate sunt necesare pentru a rula programele specificate. De exemplu, pentru a determina ceea ce aveți nevoie pentru a lansa în execuție interpretorul de comenzi Bash (<code>/bin/bash</code>), tasteți următoarele: <pre>ldd /bin/bash libtermcap.so.2 => /lib/libtermcap.so.2 (0x400003000) libc.so.6 => /lib/libc.so.6 (0x40006000) /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00000000)</pre>

less

Scop	Vizualizează fișierele text ecran cu ecran (și poate derula fișierul în sens invers)
Sintaxă	<code>less [options] filenames</code>
Opțiuni	-? afișează lista comenzilor pe care le puteți utiliza în <code>less</code> -p <i>text</i> afișează prima linie unde se găsește <i>text</i> -s reduce mai multe linii vide la una singură
Descriere	Comanda <code>less</code> afișează fișierele specificate ecran cu ecran. Spre deosebire de <code>more</code> , puteți să tasteți <code>b</code> , <code>Ctrl+B</code> sau <code>ESC+V</code> pentru a derula conținutul fișierului înapoi. Pentru a vedea comenzile pe care le puteți folosi pentru a interacționa cu <code>less</code> , apăsați <code>h</code> în timp ce vizualizați fișierul folosind comanda <code>less</code> .

ln

Scop	Creează o legătură hard sau simbolică (pseudonime) cu fișiere și directoare
Sintaxă	<code>ln[options] existing_file new_name</code>
Opțiuni	-b creează copii de siguranță ale fișierelor ce vor fi eliminate -d creează o legătură hard cu un director (numai pentru <code>root</code>) -f elimină fișierul existent care are numele <i>new_name</i> -help afișează un mesaj de asistență -s creează o legătură simbolică -v afișează rezultatul detaliat
Descriere	Comanda <code>ln</code> atribuie un nume nou unui fișier existent. Cu opțiunea <code>-s</code> puteți crea legături simbolice care pot exista între fișierele sistem. De asemenea, cu legături simbolice puteți vedea informații despre legătură, folosind comanda <code>ls -l</code> . Altfel, <code>ls -l</code> arată două fișiere distincte pentru un fișier și legătura lui hard.

locate

Scop	Listează toate fișierele ce corespund unui model specificat dintr-o bază de date actualizată periodic
Sintaxă	<code>locate pattern</code>
Opțiuni	Nici una
Descriere	Comanda <code>locate</code> caută, într-o bază de date ce conține fișiere, nume care se potrivesc cu un model specificat. Sistemul dumneavoastră Linux este setat să actualizeze periodic această bază de date. Dacă nu sunteți siguri de locația unui fișier, tasteți comanda <code>locate</code> urmată de o parte din numele fișierului. De exemplu, iată cum puteți căuta fișierul <code>XF86Config</code> : <pre>locate XF86Config /usr/X11R6/lib/X11/XF86Config /usr/X11R6/lib/X11/XF86Config.eg</pre>

lpr

Scop	Tipărește unul sau mai multe fișiere
Sintaxă	<code>lpr [options] [files]</code>
Opțiuni	-P <i>printer</i> tipărește la imprimanta specificată (numele apare în fișierul <code>/etc/printcap</code>) -# <i>N</i> (unde <i>N</i> este un număr) tipărește <i>N</i> copii ale fiecărui fișier -h suprimă pagina burst (prima pagină cu informații despre utilizator) -m trimite un mesaj la terminarea tipăririi -r elimină fișierul după tipărire -J <i>jobname</i> tipărește numele acestei sarcini pe pagina introductivă -U <i>username</i> tipărește numele acestui utilizator pe pagina introductivă

Descriere	Comanda <code>lpr</code> tipărește fișierele specificate, utilizând sistemul de derulare a tipăririi (spooling) folosit de calculatorul dumneavoastră. Dacă nu sunt specificate numele fișierelor, <code>lpr</code> citește datele de la intrarea standard. Puteți tipări la imprimanta specificată cu opțiunea <code>-P</code> .
ls	
Scop	Listează conținutul unui director
Sintaxă	<code>ls [options] [directory_name]</code>
Opțiuni	-a afișează toate fișierele, inclusiv pe cele care încep cu un punct (.) -b afișează caracterele netipăribile din numele fișierelor, utilizând baza opt -c sortează în funcție de data la care au fost create fișierele -d listează directoarele ca pe orice fișier (în loc să listeze conținutul lor) -f listează conținutul directoarelor fără să-l sorteze (exact așa cum apar acestea pe disc) -i afișează informațiile inode -l afișează fișierul în formatul lung, cu informații detaliate -p adaugă un caracter la numele fișierului, pentru a indica tipul acestuia -r sortează lista în ordine alfabetică inversă -s arată mărimea fiecărui fișier (în kilobaiți) lângă numele acestuia -t sortează listă în funcție de marca de timp a fișierului -1 afișează o listă de nume de fișiere organizată pe o coloană -R listează recursiv fișierele din toate subdirectoarele
Descriere	Comanda <code>ls</code> afișează conținutul directorului specificat. Dacă omiteți numele directorului, <code>ls</code> afișează conținutul directorului curent. În mod prestabilit, <code>ls</code> nu listează fișierele al căror nume începe cu un punct (.); pentru a vedea toate fișierele tastezi <code>ls -a</code> . Puteți vedea toate detaliile despre fișiere (inclusiv mărimea, utilizatorul, grupul proprietar și setările de acces pentru citire – scriere – execuție), utilizând comanda <code>ls -l</code> .
man	
Scop	Vizualizează paginile manualului online (denumite, de asemenea, pagini man)
Sintaxă	<code>man [options] [section] command</code>
Opțiuni	-C <i>cfile</i> specifică fișierul de configurare man (fișierul prestabilit este <code>/etc/man.config</code>) -P <i>pager</i> specifică programul ce trebuie folosit pentru afișarea pagină cu pagină (de exemplu <code>less</code>) -a afișează toate paginile man care conțin o anumită comandă -h afișează un mesaj de asistență și încheie execuția comenzii -w arată locația paginilor man ce vor fi afișate
Descriere	Comanda <code>man</code> afișează paginile man pentru comanda specificată. La fel de bine, dacă cunoașteți secțiunea pentru o pagină man, o puteți furniza. De exemplu, toate paginile <code>man Tc1/TK</code> sunt în secțiunea <code>n</code> . Astfel, puteți vizualiza pagina man pentru comanda <code>Tc1/TK pack</code> , utilizând comanda <code>man n pack</code> .
mkdir	
Scop	Creează un director
Sintaxă	<code>mkdir [options] directory_name</code>
Opțiuni	-m <i>mode</i> atribuie noului director setarea de acces specificată -p creează directoarele părinte, dacă ele nu există deja
Descriere	Comanda <code>mkdir</code> creează directorul specificat

mkfs	
Scop	Creează un fișier sistem Linux pe o partiție a unui hard-disc sau pe o dischetă
Sintaxă	<code>mkfs [-V] [-t] [options] device_name [blocks]</code>
Opțiuni	-v furnizează rezultatul detaliat, necesar pentru testare -t <i>fstype</i> specifică tipul sistemului de fișiere (cum ar fi <code>ext2</code>) -c verifică dacă dispozitivul are blocuri defecte, înainte de crearea sistemului de fișiere -l <i>filename</i> citește lista blocurilor defecte din fișierul specificat
Descriere	Comanda <code>mkfs</code> creează un sistem de fișiere Linux pe dispozitivul specificat. Dispozitivul este, de obicei, o partiție de hard-disc sau o dischetă (care a fost formatată cu <code>fdformat</code>)
mknod	
Scop	Creează un fișier dispozitiv cu numerele minore și majore specificate
Sintaxă	<code>mknod device_file (b c) major minor</code>
Opțiuni	Nici una
Descriere	Comanda <code>mknod</code> creează un fișier dispozitiv (cum ar fi cele din directorul <code>/dev</code>) prin care sistemul de operare accesează dispozitivele fizice cum ar fi hard-discul, portul serial, tastatura și mouse-ul. Pentru a crea un fișier dispozitiv trebuie să vă conectați ca <code>root</code> (rădăcină) și să cunoașteți numerele minore și majore ale dispozitivului pentru care este creat acest fișier. În plus, trebuie să specificați una din literele <code>b</code> sau <code>c</code> pentru a indica dacă dispozitivul este orientat către bloc sau caracter. De obicei, executați acest pas urmând instrucțiunile specifice dintr-un document HOWTO.
mkswap	
Scop	Creează un spațiu virtual pentru Linux
Sintaxă	<code>mkswap device_or_file numblocks</code>
Opțiuni	Nici una
Descriere	Comanda <code>mkswap</code> creează un spațiu virtual necesar pentru a fi utilizat de nucleul Linux. Dacă creați un spațiu virtual într-o partiție de disc, specificați numele dispozitivului partiției (cum ar fi <code>/dev/hda2</code>) drept al doilea argument al comenzii <code>mkswap</code> . Dacă doriți să utilizați un fișier ca spațiu virtual, creați fișierul cu o comandă cum ar fi <code>dd if=/dev/zero of=swapfile bs=1024 count=16384</code> . Apoi tastezi <code>mkswap swapfile 16384</code> pentru a crea spațiul virtual. Trebuie să utilizați comanda <code>swapon</code> pentru a activa un spațiu virtual.
more	
Scop	Vizualizează fișierele text ecran cu ecran
Sintaxă	<code>more [options] filenames</code>
Opțiuni	+<i>N</i> (unde <i>N</i> este un număr) afișează fișierul începând de la numărul de linie specificat +<i>pattern</i> afișează două linii înainte de model -s reduce mai multe linii vide la una singură
Descriere	Comanda <code>more</code> afișează fișierele specificate ecran cu ecran. Pentru a vedea comenzile pe care le puteți utiliza în <code>more</code> , apăsați <code>h</code> în timp ce vizualizați un fișier cu această comandă. Pentru o vizualizare în continuare a fișierului, utilizați comanda <code>less</code> .

mount

Scop	Asociază un dispozitiv fizic unui anumit director din fișierul sistem Linux
Sintaxă	<code>mount [options] device directory</code>
Opțiuni	-a instalează toate dispozitivele listate în fișierul <code>/etc/fstab</code> -h afișează un mesaj de asistență și încheie execuția comenzii -r montează dispozitivul numai pentru citire (scrierea nu este permisă) -t <i>fstype</i> specifică tipul fișierului sistem pe dispozitiv -v afișează mesajele detaliat -v afișează numărul versiunii și iese din comandă
Descriere	Comanda <code>mount</code> atașează conținutul unui dispozitiv fizic la un anumit director din fișierul sistem Linux. De exemplu, puteți instala un CD-ROM în directorul <code>/cdrom</code> . Apoi, puteți accesa conținutul CD-ROM-ului din directorul <code>/cdrom</code> (cu alte cuvinte, directorul rădăcină al CD-ROM-ului apare ca fiind <code>/cdrom</code> după operația de instalare). Pentru a vedea lista directorului RedHat de pe CD-ROM, ar trebui să tastați <code>ls /cdrom/RedHat</code> .

mv

Scop	Redenumeste fișiere și directoare sau le mută dintr-un director în altul
Sintaxă	<code>mv [options] source destination</code>
Opțiuni	-b creează copii de siguranță ale fișierelor care vor fi mutate sau redenumite -f elimină fișierele existente fără să emită vreun mesaj de avertizare -i solicită confirmare înainte de suprascrierea oricărui fișier existent -v afișează numele fișierului înainte de a-l muta
Descriere	Comanda <code>mv</code> redenumeste un fișier sau îl mută în alt director. Comanda lucrează atât pentru fișierele simple, cât și pentru directoare. Astfel, puteți redenumi fișierul <code>sample</code> în <code>sample.old</code> folosind comanda <code>mv sample sample.old</code> . Pe de altă parte, puteți muta fișierul <code>/tmp/sample</code> în directorul <code>/usr/local/sample</code> utilizând comanda <code>mv /tmp/sample /usr/local/sample</code> .

nice

Scop	Rulează un program la un nivel de prioritate redus sau ridicat
Sintaxă	<code>nice [options] program</code>
Opțiuni	+<i>n</i> (<i>n</i> = număr) adună <i>n</i> la valoarea nice (valorile pozitive sunt cu prioritate scăzută) -<i>n</i> (<i>n</i> = număr) scade <i>n</i> din valoarea nice (valorile negative înseamnă prioritate ridicată)
Descriere	Comanda <code>nice</code> vă permite să rulați un program cu prioritate scăzută sau ridicată. În mod prestabilit, programele rulează la valoarea nice zero. Adăugând la valoarea nice prioritatea scade, în timp ce scăzând din valoarea nice prioritatea programului crește. Numai <code>root</code> poate scădea valoarea nice.

nl

Scop	Adaugă numere liniilor de text dintr-un fișier care conțin cel puțin un caracter și scrie rezultatul la ieșirea standard
Sintaxă	<code>nl [options] [file]</code>
Opțiuni	-ba numără toate liniile -bt numără numai liniile de text (opțiune prestabilită) -sc separă textul de numerele liniilor, utilizând caracterul c (caracterul prestabilit este tabulatorul) -wn utilizează <i>n</i> coloane pentru a afișa numerele liniilor

nl

Descriere	Comanda <code>nl</code> adaugă un număr fiecărei linii de text (care conține cel puțin un caracter) dintr-un fișier și scrie rezultatul la ieșirea standard. Să presupunem fișierul <code>sample.txt</code> cu următoarele linii: A line followed by a blank line and (O linie urmată de o linie vidă și) then another non-blank line. (apoi alta linie nevidă) Aplicând comanda <code>nl</code> acestui fișier se va ajunge la următorul rezultat: <code>nl sample.txt</code> 1. A line followed by blank line and 2. then another non-blank line
------------------	--

passwd

Scop	Schimbă parola
Sintaxă	<code>passwd [username]</code>
Opțiuni	Nici una
Descriere	Comanda <code>passwd</code> schimbă parola dumneavoastră. Se solicită vechea parolă, urmată de parola nouă. Dacă vă conectați ca <code>root</code> puteți schimba parola altui utilizator prin specificarea numelui utilizatorului ca argument la comanda <code>passwd</code> .

paste

Scop	Scrie la ieșirea standard liniile corespunzător din fiecare fișier, separate de un tabulator
Sintaxă	<code>paste file1 file2 [...]</code>
Opțiuni	-s lipește toate liniile dintr-un fișier și apoi trece la lipirea celor din alt fișier, în loc să lipească câte o linie din fiecare fișier -d <i>delim</i> utilizează separatori din lista de caractere, în locul tabulatorului -determină <code>paste</code> să folosească intrarea standard ca fișier
Descriere	Comanda <code>paste</code> preia câte o linie din fiecare fișier din comandă și o scrie la ieșirea standard, folosind tabulatorul ca separator între linii. Cu opțiunea <code>-s</code> , comanda <code>paste</code> poate, de asemenea, concatena toate liniile dintr-un fișier într-o singură linie foarte mare.

patch

Scop	Aplică rezultatul comenzii <code>diff</code> unui fișier original
Sintaxă	<code>patch [options] < patch_file</code>
Opțiuni	-c determină ca fișierul <code>patch</code> să fie interpretat ca o comandă <code>diff</code> -e forțează interpretarea fișierului <code>patch</code> ca un script <code>ed</code> -f forțează fișierul <code>patch</code> să fie aplicat indiferent de consecințe -n determină ca fișierul <code>patch</code> să fie interpretat ca o comandă <code>diff</code> normală -p<i>N</i> elimină totul până la <i>N</i> caractere slash din numele căii -R arată că fișierul de corecturi a fost creat prin schimbul de date între fișiere vechi și noi -u determină ca fișierul de corecturi să fie interpretat ca o comandă <code>diff</code> unificată -v afișează numărul versiunii
Descriere	Comanda <code>patch</code> este utilizată pentru a actualiza un fișier original prin aplicarea tuturor diferențelor dintre original și versiunea revizuită. Diferențele sunt sub forma unui rezultat al comenzii <code>diff</code> , stocate în fișierul <code>patch_file</code> .

printenv

<i>Scop</i>	Vizualizează o listă de variabile de mediu
<i>Sintaxă</i>	printenv
<i>Opțiuni</i>	Nici una
<i>Descriere</i>	Comanda printenv afișează lista tuturor variabilelor de mediu curente.

ps

<i>Scop</i>	Afișează starea proceselor (programelor) care rulează în sistem
<i>Sintaxă</i>	ps [options]
<i>Opțiuni</i>	[Observație: Spre deosebire de alte comenzi, opțiunile comenzii ps nu au prefixul -] a afișează procesele altor utilizatori f afișează sub formă de arbore procesele care rulează j afișează rezultatul utilizând formatul sarcinilor l afișează rezultatul în format lung, cu multe detalii pentru fiecare proces m afișează informații despre memoria ocupată pentru fiecare proces u afișează numele utilizatorului și ora la care este lansat procesul x afișează procesele care nu sunt asociate cu vreun terminal
<i>Descriere</i>	Comanda ps afișează starea proceselor care rulează în sistem. Tastând ps fără nici o opțiune obțineți o listă a proceselor pe care le-ați lansat în execuție. Pentru a vedea o listă a tuturor proceselor din sistem, tasteți ps ax (sau ps aux, dacă doriți mai multe detalii despre fiecare proces).

ps tree

<i>Scop</i>	Afișează toate procesele care rulează sub forma unui arbore
<i>Sintaxă</i>	ps tree [options] [pid]
<i>Opțiuni</i>	-a arată argumentele liniei de comandă -c nu compactează subarborii -l afișează linii lungi -n sortează procesele după identificatorul acestora (în locul numelui) -p arată identificatorii proceselor
<i>Descriere</i>	Comanda ps tree afișează toate procesele sub formă de arbore, simplificând înțelegerea relațiilor părinte-copil dintre procese. Iată cum arată mesajul tipic afișat de comanda ps tree :

```

Pstree
init--+-atd
|   - crond
|   - gpm
|   - httpd ---10*[httpd]
|   - inetd -- in . telnetd --- bash --- pstree
|       - 2*[ in . telnetd --- bash --- su --- bash]
|   - kerneld
|   - kflushd
|   - klogd
|   - kswapd
|   - 10ve
|   - lpd

```

- lpnetd
- S*[mingetty]
- nmbd
- postmaster
- pppd
- sendmail
- smbd
- syslogd
- update

pwd

<i>Scop</i>	Afișează directorul de lucru curent
<i>Sintaxă</i>	pwd
<i>Opțiuni</i>	Nici una
<i>Descriere</i>	Comanda pwd afișează directorul de lucru curent. pwd este o comandă internă a interpretorului de comenzi Bash.

reboot

<i>Scop</i>	Încheie toate procesele și reîncarcă sistemul de operare (trebuie să fiți conectat ca root)
<i>Sintaxă</i>	reboot [options]
<i>Opțiuni</i>	-n nu salvează pe disc bufferile din memorie înainte de reinițializarea sistemului -f forțează oprirea fără apelarea comenzii /sbin/shutdown -i oprește toate interfețele de rețea înainte de reinițializarea sistemului
<i>Descriere</i>	Comanda reboot permite superutilizatorului (root) să încheie toate procesele și să reinițializeze sistemul. Comanda reboot apelează comanda /sbin/shutdown cu opțiunea -r.

rm

<i>Scop</i>	Șterge unul sau mai multe fișiere
<i>Sintaxă</i>	rm [option] files
<i>Opțiuni</i>	-f elimină fișierele fără a solicita confirmare -i solicită confirmare înainte de a elimina un fișier -r elimină recursiv atât fișierele din toate subdirectoarele, cât și din directoare -v afișează numele fiecărui fișier înainte de a-l elimina
<i>Descriere</i>	Comanda rm șterge fișierele specificate. Pentru a elimina un fișier trebuie să aveți permisiunea de a scrie în directorul ce conține fișierul.

rmdir

<i>Scop</i>	Șterge un director specificat (cu condiția ca acesta să fie gol)
<i>Sintaxă</i>	rmdir [options] directory
<i>Opțiuni</i>	-p elimină orice director părinte care se golește
<i>Descriere</i>	Comanda rmdir șterge directoarele vide. Dacă un director nu este gol, ar trebui să folosiți comanda rm -r pentru a șterge fișierele și directorul.

sed

Scop	Copiază un fișier la ieșirea standard, după editarea în conformitate cu un set de comenzi
Sintaxă	<code>sed [options] [editing_commands] [file]</code>
Opțiuni	-e 'instrucțiuni' aplică instrucțiunile de editare a fișierului -f <i>scriptfile</i> aplică comenzile de editare din fișierul <i>scriptfile</i> -n suprimă afișarea prestabilită
Descriere	Comanda <i>sed</i> este cunoscută ca <i>editor de flux</i> – copiază un fișier la ieșirea standard, în timp ce se aplică comenzile specificate de editare. Dacă nu specificați un fișier, comanda citește de la intrarea standard. Pentru a folosi editorul de flux, trebuie să învățați comenzile de editare ale acestuia, care sunt foarte asemănătoare cu cele folosite de editorul <i>ed</i> (descriș în capitolul 24).

shutdown

Scop	Încheie toate procesele și oprește (sau reinițializează) sistemul
Sintaxă	<code>shutdown [options] time [messages]</code>
Opțiuni	-t specifică timpul dintre mesaj și semnalul de terminare -h oprește sistemul după terminarea tuturor proceselor -r reîncarcă sistemul după terminarea tuturor proceselor -f realizează o reinițializare rapidă -k trimite mesaje de avertizare, dar nu realizează închiderea propriu-zisă a sistemului -c anulează oprirea sistemului aflată în curs de desfășurare
Descriere	Comanda <i>shutdown</i> (calea completă este <i>/sbin/shutdown</i>) duce la oprirea sistemului Linux în maniera obișnuită. Trebuie să specificați momentul când să înceapă oprirea; utilizați cuvântul cheie <i>now</i> pentru a opri imediat sistemul. Trebuie să fiți conectat ca <i>root</i> pentru a rula comanda <i>shutdown</i> .

sort

Scop	Sortează sau unește liniile dintr-un fișier și le scrie apoi la ieșirea standard
Sintaxă	<code>sort [options] [files]</code>
Opțiuni	-c verifică dacă fișierele sunt deja sortate, iar dacă nu sunt afișează un mesaj de eroare -m unește fișierele, sortându-le ca un grup -b ignoră spațiile libere de la început -d sortează în ordine alfabetică (utilizând numai litere, cifre și spații libere) -f tratează literele mici la fel ca majusculele echivalente -k <i>POS1[,POS2]</i> specifică câmpul de sortare prin caracterele dintre două poziții <i>POS1</i> și <i>POS2</i> -o <i>file</i> scrie în fișierul specificat în loc de ieșirea standard -r sortează în ordine inversă -g sortează numeric, dar folosește conversia la numere reale -i ignoră caracterele netipăribile -n sortează numeric (folosită atunci când fiecare linie începe cu un număr) -tC specifică caracterul separator +N consideră numai caracterele de la poziția <i>N</i> înainte (0 = prima poziție)
Descriere	Comanda <i>sort</i> sortează liniile din unul sau mai multe fișiere și le scrie la ieșirea standard. Aceași comandă poate, de asemenea, să unească câteva fișiere (când utilizați opțiunea -m) rezultând un fișier sortat și fuzionat corespunzător.

split

Scop	Împarte un fișier în câteva fișiere mai mici
Sintaxă	<code>split [options] file [prefix]</code>
Opțiuni	-l <i>N</i> (unde <i>N</i> este un număr) pune <i>N</i> linii în fiecare fișier -N (unde <i>N</i> este un număr) pune <i>N</i> linii în fiecare fișier -b <i>Nk</i> împarte fișierul la fiecare <i>N</i> kilobaiți -C <i>Nk</i> pune oricât de multe linii, fără să depășească însă, <i>N</i> kilobaiți pe fișier
Descriere	Comanda <i>split</i> împarte un fișier mare în fișiere mai mici. În mod prestabilit, <i>split</i> pune 1000 de linii în fiecare fișier. Fișierele sunt denumite prin grupuri de litere aa, ab, ac, etc. Puteți specifica un prefix pentru numele fișierelor. De exemplu, pentru a împărți o arhivă mare în fișiere mai mici, care să încapă pe câteva dischete de înaltă densitate de 3,5 inch, utilizați comanda <i>split</i> astfel: <code>split -C 1440k bigfile.tar disk.</code> Această comandă creează fișierele denumite <i>disk.aa</i> , <i>disk.ab</i> etc.

su

Scop	Vă permite să intrați în sistem folosind numele altui utilizator
Sintaxă	<code>su [options] [username]</code>
Opțiuni	-c <i>COMMAND</i> transmite comanda interpretorului de comenzi -f împiedică citirea fișierului de inițializare (<i>.cshrc</i>) când interpretorul de comenzi este <i>csh</i> sau <i>tcsh</i> -l face un shell de conectare citind fișierul de inițializare al utilizatorului -p păstrează variabilele de mediu <i>HOME</i>, <i>USER</i>, <i>LOGNAME</i> și <i>SHELL</i> -s <i>SHELL</i> rulează programul <i>SHELL</i> specificat în locul interpretorului de comenzi prestabilit al utilizatorului
Descriere	Comanda <i>su</i> vă permite să luați identitatea altui utilizator. Trebuie să furnizați parola celui utilizator înainte să puteți continua. Dacă nu furnizați numele utilizatorului, <i>su</i> consideră că doriți să intrați în sistem ca <i>root</i> (rădăcină).

swapoff

Scop	Dezactivează fișierul sau dispozitivul virtual specificat
Sintaxă	<code>swapoff device</code>
Opțiuni	Nici una
Descriere	Comanda <i>swapoff</i> oprește Linux din utilizarea dispozitivului sau fișierului specificat ca spațiu de schimb.

swapon

Scop	Activează dispozitivul sau fișierul de schimb specificat
Sintaxă	<code>swapon [-a] device</code>
Opțiuni	-a activează toate dispozitivele de schimb listate în fișierul <i>/etc/fstab</i> .
Descriere	Comanda <i>swapon</i> activează dispozitivul sau fișierul specificat ca spațiu de schimb. În timpul inițializării sistemului toate spațiile virtuale sunt activate de comanda <i>swapon -a</i> care este invocată de fișierul script <i>/etc/rc.d/rc.sysinit</i>

sync

Scop	Scrie bufferele pe disc
Sintaxă	sync
Opțiuni	Nici una
Descriere	Când nu puteți să închideți sistemul dumneavoastră Linux în maniera obișnuită (de exemplu, când nu puteți executa comenzile shutdown, halt sau reboot), ar trebui să tastați sync înainte de închiderea calculatorului.

tac

Scop	Copiază un fișier, linie cu linie, la ieșirea standard în ordine inversă (mai întâi este copiată ultima linie)
Sintaxă	tac file
Opțiuni	-b plasează separatorul la începutul fiecărei linii -r tratează șirul separator specificat prin opțiunea -s ca o expresie regulată -s sep specifică un separator (în loc de caracterul linie nouă prestabilit)
Descriere	Comanda tac afișează fișierul text specificat în ordine inversă, copiind liniile la ieșirea standard în ordine inversă. În mod prestabilit, tac tratează fiecare linie ca o înregistrare și utilizează caracterul de linie nouă ca separator de înregistrare. Totuși, puteți specifica un caracter separator diferit, caz în care tac va copia acele înregistrări la ieșirea standard în ordine inversă.

tail

Scop	Vizualizează ultimele linii dintr-un fișier
Sintaxă	tail [options] file
Opțiuni	-N (unde N este un număr) afișează ultimele N linii -nN (unde N este un număr) afișează ultimele N linii -f citește fișierul la intervale regulate de timp și afișează toate liniile noi
Descriere	Comanda tail afișează liniile de la sfârșitul fișierului specificat. În mod prestabilit, tail afișează ultimele 10 linii din fișier. Pentru a vizualiza ultimele 24 de linii dintr-un fișier denumit messages, tastați tail -24 messages.

tar

Scop	Creează o arhivă de fișiere sau extrage fișiere dintr-o arhivă
Sintaxă	tar [options] files_or_directories
Opțiuni	-c creează o arhivă nouă -d compară fișierele dintr-o arhivă cu fișierele din directorul curent -r mai adaugă la arhivă câteva fișiere -t listează conținutul unei arhive -x extrage dintr-o arhivă -C directory extrage fișierele în directorul specificat -f file utilizează fișierul specificat ca arhivă, în locul unei benzi -L n specifică faptul că banda are capacitatea de n kilobaiți -N date afișează numai fișierele create după data specificată -T file arhivează sau extrage numele fișierelor specificate în fișierul file -v afișează mesajele detaliat -z comprimă sau decomprimă arhiva folosind comanda gzip

tar

Descriere	Comanda tar creează o arhivă de fișiere sau extrage fișiere dintr-o arhivă existentă. În mod prestabilit, tar consideră că arhiva se află pe o bandă. Totuși, puteți utiliza opțiunea -f pentru a specifica un fișier drept arhivă.
------------------	---

top

Scop	Listează procesele ce rulează curent, aranjate în ordinea de partajare a timpului CPU
Sintaxă	top [q] [d delay]
Opțiuni	q determină top să ruleze cu prioritate maximă (trebuie să fiți conectat ca root) d delay specifică întârzierea dintre actualizări, în secunde
Descriere	Comanda top furnizează un ecran întreg de text cu procesele aranjate în funcție de partajarea timpului lor CPU. Prestabilit, top actualizează afișarea la fiecare cinci secunde. Apăsați q sau Ctrl+C pentru a ieși din această comandă.

touch

Scop	Modifică marca de timp a unui fișier
Sintaxă	touch [options] files
Opțiuni	-c împiedică comanda să creeze un fișier care nu există -d time utilizează timpul specificat -r file utilizează marca de timp a fișierului specificat -t MMDDhhmm[CC]YY[.ss] utilizează data și ora specificată
Descriere	Comanda touch vă permite să schimbați data și ora ultimei modificări a unui fișier (această informație este stocată cu fișierul). Dacă utilizați touch fără alte opțiuni, data și ora curentă sunt utilizate ca marcă de timp pentru fișier. Dacă fișierul specificat nu există, touch creează un nou fișier cu mărimea de 0 baiți.

tr

Scop	Copiază de la intrarea standard la ieșirea standard în timp ce substituie un set de caractere cu altul
Sintaxă	tr [options] string1 [string2]
Opțiuni	-c completează caracterele în string1 cu coduri ASCII 001-377 -d șterge de la intrare toate caracterele specificate în string1 -s înlocuiește succesiunile repetate ale oricărui caracter din string1 cu un singur caracter
Descriere	Comanda tr substituie toate caracterele de la string1 din intrare cu caracterele corespunzătoare din string2. De exemplu, pentru a converti fișierul sample.1c într-un fișier sample.1c în care toate literele sunt transformate în majuscule, ar trebui să tastați: tr [a-z] [A-Z] < sample.1c > sample.1c Pentru a înlocui aparițiile repetate de noi linii într-un fișier cu un singur caracter de linie nouă, ar trebui să tastați: tr -s '\n' <infile> outfile

tty

Scop	Afișează numele dispozitivului terminalului
Sintaxă	tty
Opțiuni	Nici una
Descriere	Comanda tty afișează numele terminalului conectat la intrarea standard. Comanda este utilizată în script-uri de interpretor care au nevoie de numele terminalului.

type	
Scop	Afișează tipul unei comenzi (dacă este o comandă de interpretor internă sau este un program executabil, separat)
Sintaxă	<code>type command</code>
Opțiuni	Nici una
Descriere	Comanda <code>type</code> vă indică tipul comenzii – dacă este o comandă interpretor internă, un pseudonim sau un program executabil. De exemplu, am utilizat <code>alias ll='ls -l'</code> pentru a defini pseudonimul <code>ll</code> , lăta ce obțin când verific tipul comenzii <code>ll</code> : <pre>type ll ll is aliased to 'ls -l'</pre>
umount	
Scop	Separă un dispozitiv din sistemul de fișiere Linux
Sintaxă	<code>umount device</code>
Opțiuni	Nici una
Descriere	Comanda <code>umount</code> elimină asocierea dintre un dispozitiv și un director în sistemul de fișiere Linux. Numai utilizatorul <code>root</code> poate executa comanda <code>umount</code> .
unalias	
Scop	Șterge un pseudonim definit anterior prin comanda <code>alias</code>
Sintaxă	<code>unalias abbreviation</code>
Opțiuni	Nici una
Descriere	Comanda <code>unalias</code> elimină un pseudonim definit anterior utilizând comanda <code>alias</code> . <code>Unalias</code> este o comandă internă a interpretorului de comenzi Bash.
uname	
Scop	Afișează informații despre sistem, cum ar fi tipul de calculator și sistemul de operare
Sintaxă	<code>uname [options]</code>
Opțiuni	-a afișează toate informațiile -m afișează tipul componentelor hardware (de exemplu: i586) -n afișează numele de gazdă al calculatorului -p afișează tipul procesorului (acesta apare ca <code>unknown</code> – necunoscut) -r afișează versiunea sistemului de operare (de exemplu 2.0.32) -s afișează numele sistemului de operare -v afișează versiunea sistemului de operare (sub forma datei la care s-a efectuat compilarea)
Descriere	Comanda <code>uname</code> afișează o varietate de informații despre calculatorul și sistemul dumneavoastră de operare (Linux).
uncompress	
Scop	Decomprimă unul sau mai multe fișiere care au fost comprimate utilizând comanda <code>compress</code>
Sintaxă	<code>uncompress [-cdrvV] files</code>

uncompress	
Opțiuni	-c scrie rezultatul la ieșirea standard și reține originalul -r decomprimă recursiv fișierele în toate subdirectoarele -v afișează un mesaj pe măsură ce este decomprimat fiecare fișier -v tipărește numărul versiunii și încheie execuția comenzii
Descriere	Comanda <code>uncompress</code> decomprimă fiecare fișier specificat și înlocuiește versiunea comprimată cu originalul (și elimină sufixul <code>.z</code> adăugat la numele fișierului comprimat). Comanda <code>uncompress</code> este identică cu comanda <code>compress</code> , urmată de opțiunea <code>-d</code> .
uniq	
Scop	Scrie toate liniile unice dintr-un fișier de intrare la ieșirea standard
Sintaxă	<code>uniq [options] file</code>
Opțiuni	-N (unde <code>N</code> este un număr) ignoră primele <code>N</code> câmpuri din fiecare linie +N (unde <code>N</code> este un număr) ignoră primele <code>N</code> caractere din fiecare linie -c scrie de câte ori a apărut fiecare linie în fișier -d scrie numai liniile duplicate -u scrie numai liniile unice (opțiune prestabilită)
Descriere	Comanda <code>uniq</code> elimină liniile duplicate dintr-un fișier de intrare și copiază liniile unice la ieșirea standard. Dacă nu specificați un fișier de intrare, <code>uniq</code> citește de la intrarea standard.
wc	
Scop	Afișează numărul de baiți, de cuvinte și de linii dintr-un fișier
Sintaxă	<code>wc [options] [files]</code>
Opțiuni	-c afișează numai numărul de baiți -w afișează numai numărul de cuvinte -l afișează numai numărul de linii
Descriere	Comanda <code>wc</code> afișează numărul de baiți, de cuvinte și de linii dintr-un fișier. Dacă nu specificați un fișier de intrare, <code>wc</code> citește de la intrarea standard.
whatis	
Scop	Caută în baza de date <code>whatis</code> (vezi comanda <code>apropos</code>) cuvinte întregi
Sintaxă	<code>whatis keyword</code>
Opțiuni	Nici una.
Descriere	Comanda <code>whatis</code> caută în baza de date <code>whatis</code> (vezi intrarea pentru <code>apropos</code>) cuvântul cheie specificat și afișează rezultatul. Sunt afișate numai cuvintele întregi care corespund criteriului de căutare.
whereis	
Scop	Găsește fișierele sursă, fișierele binare și pagina man pentru o comandă
Sintaxă	<code>whereis [options] command</code>
Opțiuni	-b caută numai fișierele binare -m caută numai paginile man -s caută numai fișierele sursă

whereis

Descriere	Comanda whereis caută în directoarele uzuale (unde sunt localizate fișierele binare, paginile man și fișierele sursă) fișierele binare, paginile man și fișierele sursă ale unei comenzi. De exemplu, iată rezultatul căutării fișierelor pentru comanda rpm : whereis rpm rpm: /bin/rpm/usr/include/rpm/usr/man/man8/rpm.8
------------------	--

which

Scop	Caută o comandă în directoarele din variabila de mediu PATH
Sintaxă	which command
Opțiuni	Nici una
Descriere	Comanda which caută în directoarele din variabila de mediu PATH fișierul care va fi executat când introduceți comanda. Aceasta este o modalitate de a verifica fișierul lansat în execuție când tastați o anumită comandă.

zcat, zless, zmore

Scop	Vizualizează conținutul unui fișier text comprimat, fără a fi necesar să-i decompri
Sintaxă	zcat filename zless filename zmore filename
Opțiuni	Nici una
Descriere	Comenzile zcat , zless și zmore lucrează în același mod ca și less , more și cat . Singura diferență este că comenzile z pot citi direct fișierele comprimate cu gzip sau compress (fără a fi necesar să le decompri în prealabil cu gunzip). Aceste comenzi sunt folosite în special pentru citirea fișierelor comprimate HOWTO din directorul /usr/doc/HOWTO .

Anexa C

Resurse Linux

Această anexă prezintă câteva resurse unde puteți găsi mai multe informații despre anumite subiecte. Cele mai multe resurse sunt pe Internet, pentru că acolo puteți găsi cele mai recente informații. Puteți descărca din Internet fișierele necesare pentru o anumită operație.

Unele resurse Internet apar în sintaxa standard a adresei URL (Uniform Resource Locator) – Capitolul 20 explică adresele URL. Dacă ați folosit un browser Web – ca de exemplu Netscape Navigator sau Microsoft Internet Explorer – sunteți probabil familiarizat deja cu adresele URL.

Pagini Web

Dacă ați navigat pe Internet, ați observat probabil că există câteva pagini Web cu informații referitoare la Linux. Un punct bun de plecare este următoarea pagină Linux:

<http://www.linuxresources.com/>

Această pagină este întreținută de SSC Inc., editorul revistei *Linux Journal*. Aici găsiți un punct de plecare pentru localizarea informațiilor despre Linux. Puteți executa clic pe butoanele mari pentru a accesa mai multe informații despre subiectele identificate prin etichetele butoanelor. În această pagină puteți găsi, de asemenea, o colecție organizată de legături cu situri Web înrudite cu Linux.

Altă sursă populară și foarte bine documentată pentru informații despre Linux este pagina inițială Linux Documentation Project (LDP) de la următoarea adresă URL:

<http://sunsite.unc.edu/LDP/>

Pe această pagină Web veți găsi foarte multe trimiteri către alte resurse Linux de pe Internet. În particular, puteți parcurge și descărca cele mai recente documente HOWTO de la adresa <http://sunsite.unc.edu/LDP/HOWTO/>.

Grupuri de discuții

Pentru a fi la curent cu cele mai recente evoluții ale sistemului Linux, aveți nevoie de acces la Internet și în special la grupuri de discuții. Puteți găsi dezbateri despre anumite subiecte în legătură cu Linux în următoarele grupuri de discuții:

- **comp.os.linux.admin** – Informații despre administrarea sistemelor Linux
- **comp.os.linux.advocacy** – Discuții despre promovarea Linux
- **comp.os.linux.announce** – Anunțuri importante despre Linux (acesta este un grup de discuții moderat, adică trebuie să trimiteți articolul moderatorului care-l afișează apoi în grupul de discuții)
- **comp.os.linux.answers** – Întrebări și răspunsuri despre Linux (toate documentele HOWTO despre Linux sunt afișate în grupuri de discuții moderate)
- **comp.os.linux.development** – Activitatea curentă de dezvoltare a sistemului Linux
- **comp.os.linux.development.apps** – Dezvoltarea aplicațiilor Linux
- **comp.os.linux.development.system** – Dezvoltarea sistemului de operare Linux
- **comp.os.linux.hardware** – Discuții despre Linux și diferite componente hardware

- `comp.os.linux.help` – Asistență pentru diferite probleme din Linux
- `comp.os.linux.misc` – Subiecte diverse despre Linux
- `comp.os.linux.networking` – Lucrul în rețea sub Linux
- `comp.os.linux.setup` – Setarea și instalarea Linux
- `comp.os.linux.x` – Discuții despre setarea și rularea sistemului X Window System sub Linux

Situri FTP cu arhive pentru Linux

Puteți descărca Linux Red Hat și alte versiuni Linux de pe unul din numeroasele situri FTP existente. Pe lângă distribuția Linux, aceste situri conțin, de asemenea, și multe pachete software ce rulează sub Linux.

Pentru a vedea cea mai recentă listă cu situri FTP ce conțin Linux Red Hat, vizitați următoarea pagină Web întreținută de Red Hat:

<http://www.redhat.com/mirrors.html>

Pentru a descărca fișiere specifice Red Hat Package Manager (RPM) pentru cea mai recentă versiune Linux Red Hat (pentru PC-uri Intel), puteți accesa situl FTP Red Hat de la următoarea adresă URL:

<ftp://ftp.redhat.com/pub/redhat/current/i386/RedHat/RPMS/>

Reviste

Linux Journal este singura revistă dedicată în întregime sistemului Linux. Pe rețeaua Web, pagina de bază a revistei se găsește la adresa <http://www.linux.journal.com/>. Iată informațiile de contact pentru această revistă:

LINUX JOURNAL

P.O. BOX 500

Missouri City, TX 77459-0500

Telefon (robot): 888-66-LINUX, 281-261-2581

FAX: 281-261-5999

E-mail: subs@ssc.com

Anexa D

Despre CD-ROM

CD-ROM-ul anexat acestei cărți conține Linux Red Hat 5.1 cu nucleul Linux 2.0.34, care este cea mai recentă versiune de Linux disponibilă (din toamna anului 1998). Linux este un sistem de operare UNIX complet pentru PC-ul dumneavoastră 486 sau Pentium. Această versiune Linux Red Hat conține mai mult de 300 MB de fișiere sursă și binare. Instalând Linux pe un PC, îl transformați într-o stație de lucru UNIX complet funcțională.



Observație

Vă rugăm să rețineți că programele incluse în acest CD-ROM sunt distribuite sub o diversitate de licențe. Unele programe (precum nucleul Linux) sunt distribuite în format sursă și binar sub incidența GNU General Public License. Câteva pachete software sunt distribuite shareware și trebuie să plătiți o taxă nominală dacă vă decideți să le folosiți. În toate cazurile, programul este protejat de legislația drepturilor de autor. După instalarea Linux Red Hat, ar trebui să consultați fișierele README.* (sau acele fișiere cu nume precum COPYING și COPYRIGHT) din diferite subdirectoare ale directorului `/usr/doc` pentru a vă informa în legătură cu condițiile prevăzute în licențele fiecărui pachet software.

Instalarea este o etapă dificilă în Linux, mai ales în cazul în care aveți un calculator compatibil IBM PC care nu a fost produs de o anumită firmă. Aveți nevoie de unele informații specifice despre componentele hardware, precum controllerul de disc, placa video și unitatea CD-ROM. Linux controlează componentele hardware prin intermediul driverelor, astfel că trebuie să vă asigurați că versiunea curentă de Linux conține driverele pentru componentele dumneavoastră. Deoarece Linux este distribuit gratuit, nu puteți pretinde suport pentru anumite componente hardware. Totuși, Linux este îmbunătățit continuu, prin colaborarea programatorilor din întreaga lume. Dacă componentele dumneavoastră hardware sunt destul de cunoscute, există o șansă destul de mare să existe deja un driver pentru acestea. În orice caz, versiunea Linux Red Hat de pe CD-ROM poate fi instalată pe o mare varietate de componente hardware, astfel că probabil poate lucra și cu perifericele incluse în sistemul dumneavoastră.

Referință încrucișată. Instrucțiunile de instalare de pe CD-ROM-ul anexat acestei cărți prezintă procedura generală de instalare și configurare pentru Linux și X Window System.



Referință încrucișată

Linux Red Hat 5.1 este o versiune completă Linux cu nucleul 2.0.34 (sistemul de operare) plus o mare varietate de programe sub Linux. În particular, veți găsi următoarele programe pe CD-ROM-ul Linux Red Hat 5.1:

- Nucleul Linux 2.0.34, cu module de driver pentru cele mai importante configurații hardware pentru PC, incluzând drivere IDE/EIDE și SCSI, dispozitive PCMCIA și CD-ROM-uri.
- Un set complet cu instrumente de instalare și configurare pentru setarea dispozitivelor (precum tastatură și mouse) și a serviciilor
- Interfață grafică utilizator bazată pe pachetul XFree86 3.3.2 cu programele de gestiune a ferestrelor `fvwm2` și `AfterStep`
- Facilități TCP/IP complete pentru Internet, rețele locale și intraneturi.
- Instrumente pentru conectarea PC-ului la furnizorul de servicii Internet, folosind protocoalele PPP, SLIP sau programele de comunicație serială stabilită prin apel telefonic.

- Suită completă de aplicații Internet, inclusiv poștă electronică (sendmail, elm, pine, mailx), mesaje (inn, tin, nn), Internet Relay Chat (ircii), telnet, FTP și NFS
- Serverul Web Apache versiunea 1.2.6., pentru a vă transforma PC-ul într-un server Web și Netscape Communicator 4.05 pentru a naviga prin Internet
- Programul Samba 1.9 LAN Manager pentru conectare la Microsoft Windows.
- Câteva editoare de text (GNU Emacs 20.2, JED, Joe, vim)
- Aplicații software pentru manipularea graficii și a imaginilor, precum XV, XPaint, Xfig, Gnuplot, Ghostscript, Ghostview, GIMP, ImageMagick și xanim
- Limbaje de programare (GNU C și C++ 2.7.2.3, Perl 5.004, Tcl/Tk 8.0.2, Python 1.5.1, GNU AWK 3.0.3) și instrumente pentru dezvoltare software (GNU Debugger 4.17, RCS 5.7, GNU Bison 1.25, flex 2.5.4a și biblioteci TIFF și JPEG)
- Suport pentru standardul industrial Executable and Linking Format (ELF) și Intel Binary Compatibility Specification (iBCS)
- Suită completă de programe utilitare standard din UNIX
- Instrumente pentru accesarea și utilizarea aplicațiilor și fișierelor DOS (DOSEMU 0.66.7, mtools 3.8)
- Aplicații software pentru typesetting și formatarea textului (groff, TeX și LaTeX)
- Jocuri precum GNU Chess, xtetris, acm, colour-yahtzee, flying, fortune-mod, mysterious, paradise, xchomp, xevil, xgalaga, xgammon, xpilot.

Index

Simboluri & numere

` (accent grav), 225-226
 { } (acolade), 223, 251-252, 253-254, 276-277
 * (asterisc), 56, 69, 74, 180-181, 181-182, 184-185, 198-199, 219-220, 221, 238-239
 - (cratimă), 93, 184-185, 206
 "... " (ghilimele), 208-209, 217, 251-252
 - (linie de pauză), 256-257, 270, 562-563, 657
 - (liniuță), 93, 184-185, 206
 _ (liniuță de subliniere), 201, 607-608
 [] (paranteze pătrate), 56, 69, 105-106, 198-199, 220-221, 221
 . (punct), 180, 181-182, 205, 207, 210-211, 269-270, 534-535, 641-642, 642-643
 ... (puncte de suspensie), 199, 216-217
 ; (punct și virgulă), 216-217, 249-250, 251-252, 547-548
 ! (semn de exclamare), 214-215, 221
 ? (semn de întrebare), 198-199, 521
 # (semnul diez), 99, 104-105, 159-160, 172-173, 214-215, 216-217, 238, 241, 251-252, 300, 382-383, 443-444, 521, 531
 : (semnul două puncte), 180-181, 192-193, 195, 260, 383-384, 641-642, 642-643, 647, 648, 650-651
 = (semnul egal), 361
 - (semnul minus), 148
 + (semnul plus), 74, 361, 412-413
 % (semnul procent), 216-217, 218, 247, 567
 @ (simbolul at), 216-217, 419-420, 437-438
 \ (simbolul backslash), 29-30, 48-49, 192-193, 221, 233-234, 238-239, 250-251, 251-252, 382-383, 419-420, 485-486
 ^ (simbolul caret), 221, 486-487
 ^ (simbolul caret), câmp, 385, 386
 \$ (simbolul dolar), 211-212, 216-217, 217, 221, 249-250, 250-251, 252, 566-567, 580-581
 | (simbolul linie de redirectare), 192-193, 221, 228, 380
 / (slash), 44, 48-49, 202-203, 221, 231-232, 233-234, 270, 510-511, 642-643, 648

~ (tilda), 205, 221, 647
 10Base-T, rețele, 448-449
 16 MB, limită, 106-107
 adăugare, încetinirea sistemului după, 283-284
 buffere, 399-400, 640-641, 645, 652-654, 656-657
 cache, 283
 descriere generală, 20
 instalare și, 20, 44, 50-51
 viteza, 283
 3Com, 27-28, 112-113, 281-282, 289-290, 446-448

A

a.out, format, 106-107
 A/, comandă, 409
 ABORT, șiruri de caractere, 485-486
 Accelerated-X, 309-310
 accent grav (`), 225-226
 Accent, plăci seriale, 422-423
 acces de la distanță, 483-491, 499
 Access (Microsoft), 541-542
 Acculogic, 23
 acolade ({}), 223, 251-252, 253-254, 276-277
 action, câmp, 173
 Adaptec, 23, 110-111, 122, 289-290, 323-326, 341, 355-356
 Add NFS, buton, 52
 Add, buton, 52, 83-84
 Add/Remove Programs, opțiune, 30-31
 Adlib, plăci de sunet, 26-27, 348-349
 administrarea sistemului, 133-134
 adrese IP. *Vezi și* IP (Internet Protocol)
 calculatoare-gazdă cu mai multe, 457-458
 clase de, 431-433
 comunicații client/server, 440-441
 descriere generală, 429-434
 transformarea numelor
 calculatoarelor-gazdă în, 69
 adrese zecimale cu punct, 431-432
 Advanced Logic, plăci video, 21
 AEP (Application Environment Profile), 127-128

aexec.bat, 195
 af, câmp, 384
 after, comandă, 268-269
 Afterstep, stil, 188-190
 alimentarea cu tensiune, 89-90, 290-292, 322, 466-467, 554-555
 All, opțiune, 535-536, 536-537
 Alliance, plăci video, 305-306
 Allied Telesis, plăci Ethernet, 27-28, 447-448
 ALPS GlidePoint, dispozitiv de indicare, 369-370
 Always, controllere, 323, 326
 AMD, plăci Ethernet, 447-448
 America Online, 498, 499
 American National Standards Institute (ANSI), 136
 American Standard Code for Information Interchange (ASCII), 184-185, 362-363, 401-402, 407, 412-413, 419-420
 AMI, 323, 325
 analiză, 249-250
 Andreessen, Marc, 526
 anonime, FTP, 53, 581, 583, 600-601, 637.
 Vezi și FTP (File Transfer Protocol)
 AnotherLevel, fișiere de configurare, 185-186
 Ansel Communications, plăci Ethernet, 27-28, 447-448
 ANSI, (American National Standards Institute), 136
 anteturi, eliminare, 387-388
 Apache HTTP, server, 58-59, 69, 95-96, 518-519, 522-523, 528-529, 541
 apache, pachet, 95-96
 apăsări de taste. *Vezi și* tastaturi
 pentru abandonarea pornirii X Window System, 163-164
 pentru comutarea între consolele virtuale, 43-44
 pentru comutarea între modurile video, 164-165
 pentru interacțiunea cu programul de instalare Red Hat, 43
 apel cu ton, 410-411
 apel invers, funcții de, 596-597
 apel prin impulsuri, 410-411
 API (Application Programming Interface)
 CSS API, 466-468
 POSIX, standarde și, 127-128
 soclu API, 438-439, 440-441
 Aplicație, nivel, 427-428, 428-429
 aplicații personale de uz curent, 541-542
 APM (Advanced Power Management), 540.
 Vezi și alimentarea cu tensiune
 Apollo Computer, 141-142
 AppClass, fișier, 181-183
 append, comandă, 258
 AppleTalk, protocol, 110
 Application Environment Profile (AEP), 127-128
 Application Programming Interface (API)
 CSS API, 466-468
 POSIX, standarde și, 127-128
 soclu API, 438-439, 440-441
 Applixware Office, suită, 129
 Apricot, plăci Ethernet, 27-28, 447-448
 arce, desenare, 620-621
 argumente prestabilite, 217
 arhitectură, 93, 140-141, 437-444
 arhive, 525-526, 579-580
 ARK Logic, 305-306
 Arnet, plăci seriale, 422-423
 ARP (Address Resolution Protocol), 445-446
 ARPA (Advanced Research Projects Agency), 429-430
 ARPAnet, 429-430
 array, comandă, 258
 ArrowButton, primitive, 601-602
 ASCII (American Standard Code for Information Interchange), 184-185, 362-363, 401-402, 407, 412-413, 419-420
 asistență
 informații de, tipărire, 477
 la instalare, 42-43
 pentru instrumente GNU, 557-559, 654-655
 asistență tehnică, 541-543
 asociere, 154. *Vezi și* bind, comandă
 asocieri de taste, 652-653, 654
 AST, plăci seriale, 422-423
 asterisc (*), 56, 69, 74, 180-181, 181-182, 184-185, 198-199, 219-220, 221, 238-239
 AT Attachment Packet Interface (ATAPI), 26-27, 42-43, 107-108, 336-338, 340-343, 347-348

AT&T (American Telephone & Telegraph), 27-28, 136, 309, 447-448
 AT, comenzi, 69, 408-413, 415
 comanda AT\$, 413
 comanda AT&C1, 421
 comanda AT&F, 411-412
 comanda AT&Fn, 410-411
 comanda AT&S0, 421
 comanda ATA, 410-411, 421
 comanda ATDL, 410-411
 comanda ATDPnumber, 410-411
 comanda ATDT, 416-417
 comanda ATDTnumber, 410-411
 comanda ATE1, 409-410, 421
 comanda ATH, 410-411
 comanda ATLn, 409-410
 comanda ATM0, 421
 comanda ATMn, 409-410
 comanda ATO, 411-412
 comanda ATQ0, 421
 comanda ATS, 411-412
 comanda ATTRIB, 238
 comanda ATV1, 421
 comanda ATX4, 421
 comanda ATXn, 410-411
 comanda ATZn, 410-411
 AT, magistrală, 19
 at, simbol (@), 216-217, 419-420, 437-438
 ATA, plăci pentru unități de disc, 468
 ATAPI (AT Attachment Packet Interface), 26-27, 42-43, 107-108, 336-338, 340-343, 347-348
 atd, serviciu, 69
 ATI, 20, 22-23, 26-27, 144-145, 283, 304-305, 305-306, 307, 310
 aumix, pachet, 58
 AuthConfig, opțiune, 536-537
 AUTOBOOT.BAT, 40
 AUTOEXEC.BAT, 38
 aw, utilitar, 128-129
 AX.25, pachet, protocol radio, 110
 aztec, driver, 346-347
 Aztech, 25-26, 26-27, 337-338, 346
B
 backslash (â), 29-30, 48-49, 192-193, 221, 233-234, 238-239, 250-251, 251-252, 382-383, 419-420, 485-486
 BACKUP, utilitar, 31
 bara de operații, 82, 187
 bara de spațiu, 43, 46, 85
 bare de derulare, 180, 181-183, 267-268, 601-602, 603-604
 Bash, interpretor (Bourne-Again Shell)
 completarea comenzii în, 198-199
 descriere generală, 128-129, 168-169, 191-230
 funcții încorporate, 212-213
 pornire și, 168
 programarea, 211-212
 .bash_logic, fișier, 168-169
 .bash_profile, fișier, 168-169, 201-202
 baterii, 115-116
 BBS (Bulletin Board Systems), 132-134, 399
 beep, comandă, 476
 Bell Technologies, plăci seriale, 422-423
 Berkeley Sockets, 131-132, 438-439
 biblioteci, partajate, 584-589
 binare (executabile), fișiere
 extensie de fișier pentru, 30-31
 FDISK.EXE, 32, 241, 243
 FIPS.EXE, 36-38
 LOADLIN.EXE, 40
 modernizări și, 106-107
 RAWRITE.EXE, 38, 39
 RESTORRB.EXE, 37
 SCSISEL.EXE, 356
 bind, comandă, 268-269, 274-277
 BIOS (Basic Input/Output Systems), 22-23, 284
 procesoare și, 280
 unități de disc și, 313-314, 321
 Bison, 667
 bitmap, 194, 196-197, 629, 729, 746-747.
 Vezi și imagini grafice
 cursoare, 613-615
 descriere generală, 140
 opțiuni, 175-176, 177-178
 biți
 adâncime, 140, 162, 163-164, 299, 635-636
 pe secundă (bps), 401-402, 407
 per pixel (bpp), 159, 162, 299, 307-309, 310
 bloc, dispozitive de tip, 366

blocări, 120, 342
 BMP, imagini, 197. *Vezi și* bitmap
 Boca, plăci seriale, 422-423
 BogoMIPS, 78, 78-79
 boot. *Vezi și* dischete de boot; reboot
 Boot Manager și, 317-318
 de pe hard-disc, cu LILO, 318-319
 memoria și, 284
 procesul, urmărire, 41-43
 rezolvarea problemelor, 330
 unități CD-ROM și, 342-343
 bootloader, 74-76. *Vezi și* LILO (Linux Loader)
 bootp, comandă, 476
 BOOTP, protocol, 476
 Bourne-Again Shell (interpretor Bash)
 completarea comenzilor în, 198-199
 descriere generală, 128-129, 168-169, 191-230
 funcții înglobate, 212-213
 pornire și, 168
 programare, 211-212
 bps (biți pe secundă), 401-402, 407
 br, câmp, 384
 Bradley, John, 130-131
 break, comandă, 258, 476
 browsere, 131-132, 438-439
 browsere de Web, 131-132, 438-439
 browsere de Web Netscape, 524-528, 539, 667
 descărcare, 524-528
 descriere generală, 518-519
 HTTP și, 521-524
 descărcare, 524-528
 descriere generală, 518-519
 HTTP și, 521-524
 Netscape, pentru Web, 524-528, 539, 667
 BSD (Berkeley Software Distribution) UNIX, 113-114, 126, 129, 373-374, 377-378, 391
 buffere, 399-400, 640-641, 645, 652-654, 656-657
 burst, pagina de antet de, 387-388
 BusLogic, 24, 323, 325
 buton de comandă, 266-267, 267-268, 274-275
 ButtonPress, evenimente, 598

C

c, opțiune, 582
 C, programare în, 131-132, 553-554
 compilatoare, 557, 557-558, 561-564
 interpretor Bash și, 213, 214-215, 219-220, 223
 Tcl/Tk, 264-265
 tipărire și, 389-390
 C++, programare în, 59, 213, 389-390, 557-558, 562-563
 c89, compilator, 128-129
 Cabletron, plăci Ethernet, 27-28, 447-449
 cablu subțire (thinwire), 445-446, 446-447
 cabluri
 modemuri, 405-407, 421-423
 probleme cu, 324
 cache, memorie, 283
 cadre, 267-268, 427-428, 445, 603
 Caldera, sit Web
 Cancel, buton, 52
 canvas, comandă, 267-268
 capacitate de autotestare, 332, 450-452
 capete, 313-315
 caracter de tabulare pe orizontală (ât), 250-251
 caracter(e). *Vezi și* simboluri
 dispozitive, 366
 în rapoarte de verificare RPM, 96-97
 caractere (listate după nume). *Vezi și* simboluri
 caractere de înlocuire, 198-199, 238-239
 caractere escape, 412-413, 419-420, 485-487, 653-654
 caracterul backspace (\b), 250-251, 412-413, 419-420
 caracterul bell (\a), 250-251
 caracterul carriage return - retur de car (\r), 194-195, 250-251, 412-413, 419-420, 486-487
 caracterul form feed - salt la pagină nouă (\f), 250-251, 387-388, 419-420
 caracterul newline - salt la linie nouă, 194-195, 392-394, 412-413, 419-420, 486-487
 caracterul null, 486-487
 caracterul spațiu, 486-487

caracterul spațiu unic, 419-420
 caracterul tabulator, 419-420, 486-487
 caracterul tabulator orizontal (\t), 250-251
 caracterul tabulator vertical (\v), 250-251
 caracterul XOFF, 407
 caracterul XON, 407
 caractere de înlocuire, 198-199, 238-239
 caracterul backspace (\b), 250-251, 412-413, 419-420
 caracterul bell (\a), 250-251
 caracterul retur de car (\r), 194-195, 250-251, 412-413, 419-420, 486-487
 Card Services (PCMCIA Card Services for Linux), 464, 466-468
 CardBus, 466-467
 cardctl, 467-468
 caret (^), 221, 486-487
 Carnegie Mellon University, 515-516
 CascadeButton, funcție, 601-602
 case, structuri de control, 212-213
 catch, comandă, 258
 căutare și înlocuire, acțiuni de, 651, 657-658
 câmp(uri). *Vezi și* câmpuri (listate după nume)
 câmpul acțiune, 384
 câmpul af, 384
 câmpul br, 384
 câmpul cf, 384
 câmpul df, 384
 câmpul fc, 384
 câmpul ff, 384
 câmpul filtru de intrare, 386-387
 câmpul fo, 384
 câmpul gf, 384
 câmpul hl, 384
 câmpul ic, 384
 câmpul if, 386, 389-390
 câmpul lf, 385, 387-388
 câmpul lo, 385
 câmpul lp, 385, 386, 386-387
 câmpul mx, 385, 386, 387-388
 câmpul nf, 385
 câmpul niveluri de rulare, 172-173
 câmpul of, 385
 câmpul pc, 385
 câmpul pl, 385
 câmpul proces, 173
 câmpul pw, 385
 câmpul px, 385
 câmpul py, 385
 câmpul rf, 385
 câmpul rm, 385, 386, 386-387
 câmpul rp, 386, 386-387
 câmpul sb, 385
 câmpul sc, 385
 câmpul sd, 385, 386
 câmpul sf, 385, 386, 387-388
 câmpul sh, 385, 386, 387-388
 câmpul st, 385
 câmpul tf, 385
 câmpul tr, 385
 câmpul vf, 385
 câmpul filtru de intrare, 386-387
 câmpuri logice, 383-384, 386, 387-388
 CCR1, registru de configurare, 284
 cd, comandă, 101, 191-192, 205, 208, 258
 CDE (Common Desktop Environment), 600-601
 cdp, pachet, 57
 ceas
 cipuri, 296, 309
 CMOS, ceas de timp real, 22-23
 probare, 157, 158-159
 viteze, 280, 282
 cercuri, desenare, 620-621
 cereri de întrerupere (IRQ), 288-292, 318-319, 333, 349, 354, 414
 Ethernet și, 450-452
 modemuri și, 402-403
 mouse și, 365
 CERT (Computer Emergency Response Team), sit Web, 515-516
 cf, câmp, 384
 CGI (Common Gateway Interface), 529, 533-534
 chat, 485-487
 chatkey, comandă, 476
 Check for bad blocks during format, opțiune, 55
 checkbutton, comandă, 267-268
 Chips & Technology, 291-292, 305, 310, 365
 chmod, comandă, 194-195, 206, 209-210, 344
 CHS, adresare, 313-314, 321

ci, comandă, 578-579, 581-582
 CIAC (Computer Incident Advisory Capability), 515-516
 cicluri
 cicluri for, 212-213, 215-216, 225, 253-257
 cicluri while, 222-225
 cilindri, 313-315, 321
 Cirrus Logic, 291-292, 305-306, 306-307
 CIS (Card Information Structure), 465-466, 466-467
 clase
 adrese IP, 431-433
 clasa VT100, 180, 180-181
 client/server, arhitectura, 140-141, 437-444
 CLOCAL, indicator, 422-423
 Clocks, linia, 157-159
 close, comandă, 258
 cmd, comandă, 238
 CMOS, ceas de timp real, 22-23
 CMOS, memorie, 22-23, 342
 cmp, operator, 218-219
 co, comandă, 578-579, 580-581, 581-582
 co, utilitar, 578-579, 580-581, 581-582
 Cogent, plăci Ethernet, 27-28, 447-448
 COM, extensie de fișier, 30-31
 COM1 (port serial), 365, 372-373, 402-404, 414, 421
 COM2 (port serial), 365, 372-373, 402-404
 COM3 (port serial), 402-404
 COM4, 402-404
 COM4 (port serial), 402-404
 imprimante și, 372, 372-373
 modemuri și, 399, 399-403, 414, 417-419, 421-423
 modernizări Linux și, 114-115
 mouse și, 364, 367-368, 369-370
 comandă. *Vezi și* comenzi (listate după nume)
 blocuri de, 253-254
 combinații, 192-193
 completare, în Bash, 198-199
 descriere generală, 191-230
 editare, 200-201
 execuție, 225-228, 264
 istoric, 200
 în fundal, 197-199
 linie de, opțiuni, 179, 182-185, 191-193
 mod, 408, 641-642
 sintaxa, 191-202
 substituție, 250-251
 comentarii, 172-173, 251-252
 comenzi (listate după nume). *Vezi și* comandă
 comandă A/, 409
 comandă after, 268-269
 comandă append, 258
 comandă array, 258
 comandă AT\$, 413
 comandă AT&C1, 421
 comandă AT&F, 411-412
 comandă AT&Fn, 410-411
 comandă AT&S0, 421
 comandă ATA, 410-411, 421
 comandă ATDL, 410-411
 comandă ATDPnumber, 410-411
 comandă ATDT, 416-417
 comandă ATDTnumber, 410-411
 comandă ATE1, 409-410, 421
 comandă ATH, 410-411
 comandă ATLn, 409-410
 comandă ATM0, 421
 comandă ATMn, 409-410
 comandă ATO, 411-412
 comandă ATQ0, 421
 comandă ATS, 411-412
 comandă ATTRIB, 238
 comandă ATV1, 421
 comandă ATX4, 421
 comandă ATXn, 410-411
 comandă ATZn, 410-411
 comandă beep, 476
 comandă bind, 268-269, 274-277
 comandă bootp, 476
 comandă break, 258, 475
 comandă button, 266-267, 267-268, 274-275
 comandă canvas, 267-268
 comandă catch, 258
 comandă cd, 101, 191-192, 205, 208, 238, 258
 comandă chatkey, 476
 comandă checkbutton, 267-268
 comandă chmod, 194-195, 206, 209-210, 344
 comandă ci, 578-579, 581-582

comandă close, 258
 comandă cmd, 238
 comandă co, 578-579, 580-581, 581-582
 comandă compare, 261
 comandă concat, 258
 comandă config, 476
 comandă continue, 258
 comandă COPY, 238, 238-239
 comandă cp, 191-192, 211-212, 233-235, 238-239
 comandă databits, 477
 comandă dec, 477
 comandă default, 477, 480-481
 comandă DEL, 238
 comandă destroy, 268-269
 comandă dial, 477
 comandă dir, 202
 comandă DIR, 238, 238-239
 comandă E, 409
 comandă entry, 267-268
 comandă eof, 258
 comandă ERASE, 238
 comandă error, 258
 comandă eval, 258
 comandă exec, 258, 264
 comandă exit, 258, 274-275, 477
 comandă expr, 212-213, 258
 comandă fdformat, 239
 comandă file, 258
 comandă find, 208-209
 comandă flush, 258, 477
 comandă focus, 268-269
 comandă for, 255
 comandă foreach, 225, 255, 258
 comandă FORMAT, 30-31, 31-32, 36-37, 238, 242, 258
 comandă frame, 267-268
 comandă get, 101, 258, 477
 comandă GET, 522, 522-523
 comandă gets, 262-263, 263-264
 comandă glob, 258, 263-264, 264
 comandă global, 257, 258
 comandă goto, 477
 comandă gpm, 145-146, 369-370
 comandă grab, 268-269
 comandă grep, 93-94, 101-102, 122, 192-193, 199, 219-221, 231-232
 comandă gs, 397
 comandă gunzip, 99, 133-134
 comandă gzip, 99, 667, 671-672
 comandă HEAD, 529
 comandă help, 477
 comandă history, 200, 258
 comandă if, 253, 259, 261
 comandă ifconfig, 460-461, 483, 489
 comandă inc, 477
 comandă incr, 259
 comandă info, 259
 comandă init, 170-175, 196-197, 236, 417-419
 comandă join, 259, 260, 261
 comandă k, 104
 comandă kill, 163-164, 197-198
 comandă label, 267-268
 comandă LABEL, 238
 comandă lappend, 259
 comandă less, 208, 241
 comandă lindex, 259
 comandă linsert, 259
 comandă list, 259
 comandă listbox, 267-268
 comandă llength, 259
 comandă lower, 268-269
 comandă lpc, 377-378, 378-379
 comandă lpc status, 376-377
 comandă lpd, 377-378, 380, 381-382, 386-387, 387-388, 390, 392-393
 comandă lpq, 374-375, 376-377, 377-378
 comandă lpr, 376-377, 377-378, 380-381, 386, 391-392, 394-395
 comandă lprm, 376-377, 377-378
 comandă lps, 373-375
 comandă lps status, 374-376
 comandă lrange, 259
 comandă lreplace, 259
 comandă ls, 191-192, 202, 205-207, 236, 264, 339-340
 comandă lsearch, 259
 comandă m, 316-317
 comandă mail, 502-503
 comandă make boot, 117
 comandă make clean, 117
 comandă make config, 104, 105-106, 493
 comandă make dep, 117

comanda make xconfig, 104
 comanda man, 662
 comanda man bash, 212-213
 comanda mattrib, 238
 comanda mcopy, 236, 238
 comanda MD, 238
 comanda mdel, 238
 comanda mdir, 236, 238, 238-239
 comanda menu, 267-268
 comanda menubutton, 267-268
 comanda message, 267-268
 comanda mformat, 238, 239
 comanda mkdir, 129
 comanda MKDIR, 238
 comanda mke2fs, 320
 comanda mkinitrd, 120, 122
 comanda mlablel, 238
 comanda mmd, 238
 comanda modem, 478
 comanda more, 208
 comanda mount, 98, 232-233, 233-234, 235, 239-240
 comanda mrd, 238
 comanda mread, 238-239
 comanda mren, 238-239
 comanda mtype, 238-239
 comanda mv, 207
 comanda mwrite, 238-239
 comanda netmask, 478
 comanda netstat, 461-462
 comanda open, 259, 262-263
 comanda p, 316-317
 comanda pack, 268-269, 270-272
 comanda parity, 478
 comanda patch, 101, 667, 675-676
 comanda pid, 259
 comanda ping, 489-490
 comanda place, 268-269, 270-271, 273-274, 274-275
 comanda port, 478
 comanda POST, 522-523
 comanda ppl, 480-481
 comanda print, 228
 comanda proc, 257, 259
 comanda proxyarp, 478
 comanda ps, 196-198, 390, 443-444
 comanda puts, 259, 262-263
 comanda pwd, 191-192, 204-205, 259
 comanda quit, 478
 comanda QUIT, 504-505
 comanda radiobutton, 267-268
 comanda raise, 268-269
 comanda RCPT TO:, 504-505
 comanda rcs, 582
 comanda rdev, 79-80
 comanda read, 211-212, 259, 263-264
 comanda regexp, 259, 260
 comanda regsub, 259, 260
 comanda REN, 238-239
 comanda rename, 259
 comanda RENAME, 238-239
 comanda reset, 478
 comanda return, 259
 comanda rlog, 582, 583
 comanda rm, 202, 207-208
 comanda rmdir, 208
 comanda route, 489-490, 493
 comanda rpm, 92, 94-95, 96, 120, 133-134, 237, 241
 comanda scale, 267-268
 comanda scan, 259
 comanda scrollbar, 267-268
 comanda securid, 478
 comanda securidf, 478
 comanda seek, 259
 comanda selection, 268-269
 comanda send, 268-269, 478, 480-481
 comanda set, 249-250, 259
 comanda showkey, 361
 comanda shutdown, 40, 79-80, 89-90, 134-135, 322
 comanda skey, 478
 comanda sleep, 478
 comanda source, 259
 comanda speed, 478
 comanda split, 259, 260, 261, 263-264
 comanda startx, 168, 196-197, 369-370
 comanda stopbits, 478
 comanda string, 260, 261
 comanda stty, 417-419, 419-420
 comanda switch, 256-257, 260
 comanda tell, 260
 comanda term, 478, 479-480
 comanda test, 212-213

comanda text, 267-268
 comanda time, 260
 comanda timeout, 479
 comanda tk, 268-269
 comanda tkerroe, 268-269
 comanda tkwait, 268-269
 comanda toplevel, 267-268
 comanda tr, 194-195
 comanda trace, 260
 comanda TYPE, 238-239
 comanda uname, 98
 comanda unknown, 260
 comanda unmount, 232-233, 235, 236
 comanda unset, 260
 comanda update, 268-269
 comanda uplevel, 260
 comanda upvar, 260
 comanda V, 409
 comanda wait, 479
 comanda which, 213
 comanda while, 253-257, 260
 comanda winfo, 268-269
 comanda wm, 268-269
 comanda zcat, 85
 comenzi de control, 248-249, 253-254
 Command, primitivă, 603
 COMMAND.COM, 191-192
 Common Gateway Interface (CGI), 529, 533-534
 comp.os.linux.announce, grup de discuții, 98, 506-507, 510-511, 511-512, 514-515
 comp.os.linux.setup, grup de discuții, 98
 Compaq, 20, 305, 310
 compilatoare
 compilatoare C, 93-94, 136, 541, 557, 557-558, 559-564
 compilator c89, 128-129
 compilator gcc, 93-94, 136, 541, 559-564
 descriere generală, 541
 comprimare, date, 117, 525-526
 comprimarea datelor, 117, 525-526
 CompuServe, 476, 580, 581. *Vezi și* ISP (Internet Service Providers)
 Computer Systems Laboratory (NIST), 126
 computer, utilizarea termenului, 279
 Computone, plăci seriale, 423

comunicativ, mod, 409-410
 concat, comandă, 258
 conectarea comenzilor, 208-209
 conexiuni dedicate, 137
 config, comandă, 477
 .config, fișier, 104-105
 CONFIG.SYS, 38
 CONFIG_APM, opțiune, 115-116
 CONFIG_ATIXL_BUSMOUSE, opțiune, 115-116
 CONFIG_BLK_DEV_FD, opțiune, 107-108
 CONFIG_BLK_DEV_HD_IDE, opțiune, 107-108
 CONFIG_BLK_DEV_IDECD, opțiune, 107-108
 CONFIG_BLK_DEV_IDEFLOPPY, opțiune, 107-108
 CONFIG_BLK_DEV_IDESCSI, opțiune, 108-109
 CONFIG_BLK_DEV_IDETAPE, opțiune, 107-108
 CONFIG_BLK_DEV_IDE_PCMCIA, opțiune, 108-109
 CONFIG_BLK_DEV_LOOK, opțiune, 108-109
 CONFIG_BLK_DEV_LRAM, opțiune, 108-109
 CONFIG_BLK_DEV_MD, opțiune, 108-109
 CONFIG_BLK_DEV_SD, opțiune, 110
 CONFIG_BLK_DEV_TRITON, opțiune, 108-109
 CONFIG_BUSMOUSE, opțiune, 115-116
 CONFIG_EXT2_FS, opțiune, 113
 CONFIG_EX_FS, opțiune, 113
 CONFIG_FAT_FS, opțiune, 113
 CONFIG_FIREWALL, opțiune, 109-110
 CONFIG_FTAPE, opțiune, 115-116
 CONFIG_HPRS_FS, opțiune, 113-114
 CONFIG_IDE_CHIPSETS, opțiune, 108-109
 CONFIG_INET, opțiune, 109-110
 CONFIG_IPX, opțiune, 110
 CONFIG_IP_ACCT, opțiune, 109-110
 CONFIG_IP_FORWARD, opțiune, 109-110
 CONFIG_IP_MULTICAST, opțiune, 109-110
 CONFIG_IP_ROUTER, opțiune, 109-110
 CONFIG_ISO9660_FS, opțiune, 113-114

CONFIG_MATH_EMULATION, opțiune, 104-105
 CONFIG_MAX_16M, opțiune, 106-107
 CONFIG_MINIX_FX, opțiune, 113
 CONFIG_MSDOS_FS, opțiune, 113-114
 CONFIG_MS_BUSMOUSE, opțiune, 115-116
 CONFIG_NET, opțiune, 106-107
 CONFIG_NET_IPIP, opțiune, 109-110
 CONFIG_NFS_FS, opțiune, 113-114
 CONFIG_PCI, opțiune, 104-105
 CONFIG_PRINTER, opțiune, 114-115
 CONFIG_PROC_FS, opțiune, 113-114
 CONFIG_PSMOUSE, opțiune, 115-116
 CONFIG_QICO2_TAPE, opțiune, 115-116
 CONFIG_QUOTA, opțiune, 113
 CONFIG_ROOT_FS, opțiune, 113-114
 CONFIG_RTC, opțiune, 115-116
 CONFIG_SCSI, opțiune, 110
 CONFIG_SCSI_GDTH, opțiune, 111
 CONFIG_SMB_WIN95, opțiune, 113-114
 CONFIG_SYN_COOKIES, opțiune, 109-110
 CONFIG_SYSVIPC, opțiune, 106-107
 CONFIG_SYSV_FS, opțiune, 113-114
 CONFIG_UFS_FS, opțiune, 113-114
 CONFIG_UMISC, opțiune, 115-116
 CONFIG_UMSDOS_FS, opțiune, 113-114
 CONFIG_VFAT_FS, opțiune, 113-114
 CONFIG_WATCHDOG, opțiune, 115-116
 CONFIG_XIA_FS, opțiune, 113
 connect, opțiune, 487
 connect.dip, 480-483
 console
 console virtuale, 42-44
 descriere generală, 168-169
 login de la, 168-169
 pachete multimedia, 57
 pachete pentru jocuri, 57
 consultanți, 555-556
 context grafic, 609, 614-624, 636-637, 637-638
 continue, comandă, 258
 Control Panel, 30-31
 controllere de disc, 15-16, 312-314
 controlul furcii, 410-411
 conversie adresă, 321
 copiere
 blocuri de text, 659-660
 fișiere, 30-31, 233-235, 236, 238-239
 copii de siguranță, 31, 36, 134-135
 coprocesor matematic (procesor în virgulă mobilă), 104-105, 106-107, 280
 COPY, comandă, 238, 238-239
 Core, primitive, 598-600
 Corel Corporation, 129
 cos, funcție, 253
 count, variabilă, 249-251
 courier, font, 204. *Vezi și* fonturi
 cp, comandă, 191-192, 211-212, 233-235, 238-239
 CPAN (Comprehensive Perl Archive Network), 214
 CPU (unitate centrală de procesare)
 convenții de nume pentru, 286-287
 descriere generală, 279-293
 instalare și, 14-15, 19, 44, 78-79
 Interpretor Bash și, 196-198
 modernizări Linux și, 93, 104-105, 107-108
 MS-DOS și, 239-240
 tipuri de, 279-293
 Creative Labs, 25-26, 337-338, 356
 crond, serviciu, 69
 crontab, 69
 Crosstalk, 399-400
 CRTSCTS, indicator, 419-420
 crtscts, opțiune, 487
 Crystal, plăci de sunet, 26-27
 CSLIP (Compressed SLIP), 474-475, 478
 CSMA/CD (Carrier Sense Multiple Access with Collision Detection), 427-428, 444-445
 CSS (Card and Socket Services), specificație, 465-466, 466-468
 CSS API, 466-468
 CStext, funcție, 601-602
 CSU/DSU, 554-555
 CTS (Clear to Send), semnal, 404, 407, 417-419, 487
 culoare, 519-520, 628-638
 culoare de fundal, 198, 202-204. *Vezi și* culoare
 256 culori, mod, 140, 158, 178-179, 291-292, 295-296

de afișare pe ecran, 294-296
 de fundal, 179, 182-185
 format hexazecimal pentru, 176-177
 hărți de, 295-296, 610, 611-612, 612-613, 629-635
 palete de, 295-296
 pentru cursoare, 178-179, 613-615
 pentru imagini bitmap, 140
 resurse și, 179, 182-183
 RGB, mod, 140, 159-161, 176-178, 184-185, 295-296, 310, 598-599
 Current Disk Partitions, lista, 52-53
 cursă inversă pe verticală, 294-296
 cursoare
 culoare, 178-179, 613-615
 descriere generală, 609
 formă, 178-179, 613-615
 mutare, 43, 45
 nume, 178-179
 cursor_name, opțiune, 178-179
 cuvinte-cheie
 identificare RCS, 581-582, 583-584
 local, 169-170
 next, 225
 stderr, 262-263
 stdin, 262-263
 stdout, 262-263
 Cyrix 6x86, sistem, 284
D
 d, opțiune, 194-195
 D-Link, plăci Ethernet, 27-28
 daemon
 descriere generală, 105-106, 375-376
 e-mail, 501-502
 httpd (HTTP daemon), 69, 528-529, 531-533
 Point-to-Point Protocol, 484
 servere independente, și, 444-445
 tipărire și, 375-376, 390
 Danpex, plăci Ethernet, 27-28, 447-448
 DARPA (Defense Advanced Research Projects Agency), 429-430
 databits, comandă, 477
 datagrame, 439
 DataLink, nivel, 427-428, 428-429
 DCE (Data Communication Equipment), 403-404, 405-406, 406-417
 DEC (Digital Equipment Corporation), 20, 27-28, 135, 141-142, 280, 445, 447-448
 dec, comandă, 477
 decupare și lipire, 650-651
 default, comandă, 477, 480-481
 defaultroute, opțiune, 487
 DEFRAG, 36-37
 Defragment Now, buton, 37
 DEL, comandă, 238
 Delete, buton, 52
 delete, funcție, 218
 Dell Computer, 161, 285-286
 Dennis, David, 554-555
 DEO (Extended Data Out) DRAM, 283
 derularea textului, comenzi pentru, 761. *Vezi și* bare de derulare
 derularea tipăririi (spooling), 373-374, 376-378, 380-381, 381-382
 desenabile, 610
 destroy, comandă, 268-269
 dezinștalare programe, 92
 dezvoltare, software, 58-59, 135-136, 55, 557-590
 df, câmp, 384
 dial, comandă, 477
 Diamond SpeedStar, 143
 Diamond Stealth, 143, 144-145
 Diamond Viper, 20, 283, 306-307
 diff, utilitar, 582
 difuzoare, volum, 409-410
 Digi, plăci seriale, 423
 dimensiunile hârtiei, 71-73
 DIN, conector, 365
 dip (Dial-up Protocol Driver), program, 475-483, 484, 667
 dir, comandă, 238-239
 directoare
 asociere, cu puncte de montare, 52
 căi pentru, în URL-uri, 520-521
 comutarea între, 101
 creare, 98, 129
 curente, 204-205
 descriere generală, 29-30
 lista de, 205-207

manipulare, 208
mutare, 117
navigare prin, 204-205
nume de, separatori între, 48-49
permisiuni pentru, 205-207
personale, 196, 204-205
puncte de montare și, 50
rădăcină, 44, 123, 202-203, 204-207, 208-209
servere Web și, 531-532, 534-535, 535-536
sistem, 202-204
structura de, 202-205
dischete de boot, 19, 107. *Vezi și* boot
creare, 17, 31-32, 37-39, 74-75, 79-81
descriere generală, 17, 31-32
MS-DOS, 31-32, 38
pseudonime și, 201
repartiționare și, 30-31
discuri de instalare suplimentare, 39-40, 45
discuri, inițializare, 28-29, 103
creare, 17, 31-32, 37-39, 74-75, 79-81
descriere generală, 17, 31-32
MS-DOS, 31-32, 38
pseudonime și, 201
repartiționare și, 30-31
Disk Druid, 17, 48, 51-55, 231
Disk Manager, 322
Disk Setup, casetă de dialog, 51
display cu baleiaj continuu, 294-296
display, opțiune, 183-184
Display, structură, 622-623
DISPLAY, variabilă de mediu, 196
dispozitiv(e)
erori, 332, 369-370
fișiere, descriere generală, 48-49
dispozitive de indicare, 175, 409. *Vezi și* mouse
DIX, standard, 445
DMA (Direct Memory Access), 288-289, 291-292, 318-319, 332, 349, 354, 509-510
DNS (Domain Name Service), 53, 66, 495, 503-505. *Vezi și* nume de domenii
do-while, instrucțiune, 215-216
documentație, on-line, 84-90
Done, buton, 83-84
DontZap, indicator, 160-161
DontZoom, indicator, 160-161
DOS (Disk Operating System)
„secrete“, 231-245
BACKUP, utilitar, 31
comenzi, 30-31
conexiune la, întreținere, 135
emulator (DOSEMU), 57, 135, 239-245, 667, 669-670
fișiere batch (de comenzi), 195
fișiere, montare, 134-135
instalare și, 14-16, 22-23, 28-41, 52-54, 77
interpretor Bash și, 191-192, 194-195, 197-199, 202-203
LILO și, 317-318
lucrul în, înainte de instalarea Linux, 29-31
mod, repornire în, 38, 41
modemuri și, 399-400, 402-404, 421-422
modernizări Linux și, 105-106, 113-114, 118
nume de fișiere, 30-31, 48
partiții, 28-31, 52-54, 77, 231-236
plăci de sunet și, 355
procesoare și, 279
programe shell (interpretor) și, 194-195
repartiționare sub, 31-38
sisteme de fișiere, 231-236
structura de directoare, 202-203
tastaturi și, 357
tipărire și, 376-377, 392-393
TSR (programe rezidente), 105-106, 129
dos2unix, fișier, 194-195
DOSEMU (emulator), 57, 135, 239-245, 667, 669-670
două puncte (:), 180-181, 192-193, 195, 260, 383-384, 641-642, 642-643, 647, 648, 650-651
DPT, controllere, 323, 326
drawing, primitivă, 598
DrawingArea, primitivă, 603
DrawnButton, funcție, 601-602
dreptunghiuri, desenare/umplere, 619-620
drivere
descriere generală, 17
modernizări Linux și, 103, 104-117
pentru plăci seriale multiport, 423-424
DSR (Data Set Ready), semnal, 405-406

DTC, controllere, 325
DTE (Data Terminal Equipment), 403-404, 405-407, 408
DTR (Data Terminal Ready), semnal, 405-406
duplex, mod, 403-404
E
E, comandă, 409
e, opțiune, 214-215
e-mail, 57, 531, 544
adrese, 501-502
descriere generală, 136, 499
setarea calculatorului-gazdă și, 498-499
software, instalare, 499-506
each, funcție, 218
echilibrare a încărcării pe liniile seriale, 112
ECHO-PSS, plăci de sunet, 26-27, 348-349
ecou, comenzi de, 409-410, 477
ecran(e). *Vezi și* monitoare
acoperite cu fosfor, 140, 294-295
comutarea între, 43
culori, 176-178
descriere generală, 294-295
imagine de fundal pentru, 177-178
LCD (afișare cu cristale lichide), 291-292, 294-295
moduri, 163-165
suprafețe de lucru virtuale și, 185-190
terminale virtuale și, 197-199
ed, editor de text, 640-645, 652-653
Edit New Partition, casetă de dialog, 52-54
Edit User Definition, casetă de dialog, 83-84
Edit, buton, 52
editoare de text
editorul de text ed, 640-645, 652-653
editorul de text Joe, 668-669
editorul de text vi, 128-129, 195, 200-201, 211-212, 214-215, 513, 557-558, 640-641, 645-651, 652-653
editorul de text Vim, 668-669
efect de scară, 392-394
EGA (Enhanced Graphics Adapter), plăci video, 20
EIDE (Enhanced Integrated Drive Electronics), 19, 25-26, 282, 312-313, 315-316, 321
ELF (Executable and Linking Format), 107-108, 541, 557, 584-586
elipse, desenare, 620-621
elm, pachet, 57, 500-501, 503-504, 667
else, clauze, 223
elsif, clauze, 223-224
emacs, editor (GNU), 17, 58-59, 136, 200-201, 214-215, 541, 557-558, 559-560, 651-660, 667, 668
Enhanced Integrated Drive Electronics (EIDE), 19, 25-26, 282, 312-313, 315-316, 321
Ensoniq SoundScape, plăci de sunet, 26-27, 348-349, 350-351
Enter, tastă, 43
entry, comandă, 267-268
cof, comandă, 258
eq, operator, 218-219
ERASE, comandă, 238
erori
„non-System disk or disk error“, mesaj, 31
comanda mount și, 232-233
inoduri și, 322
înregistrarea în jurnal, 69-70, 388-389, 531-532
modernizări Linux și, 94-95, 97, 102, 120
remedii și, 97
RPM și, 94-95
SCSI și, 332
servere Web și, 531-532, 534-535, 535-536
structuri de control și, 212-213
verificare hard-disc pentru depistarea, 37
error, comandă, 258
ERRORS.TXT, 37
escape, caractere, 412-413, 419-420, 485-487, 653-654
ESDI (Enhanced Small Device Interface), controllere, 312-313
ESS Technologies, 27-28
/etc/dosmu.conf, 239-240, 241-242, 244-245
/etc/gettydefs, 417-420
/etc/host.conf, 134-135, 437-438, 457-458
/etc/HOSTNAME, 134-135, 457
/etc/hosts, 134-135, 441-442, 456, 457, 549
/etc/hosts.allow, 134-135, 457-458
/etc/hosts.deny, 134-135, 459

- /etc/httpd/conf/access.conf, 529-530
- /etc/httpd/conf/httpd.conf, 530-531
- /etc/httpd/conf/srm.conf, 529-530, 532-536
- /etc/inetd.conf, 442-444
- /etc/inittab, 170-175, 421-423, 459
- /etc/lilo.conf, 118-119, 122-123, 284, 318-319, 452-453
- /etc/minicom.users, 415-416
- /etc/mtools.conf, 237-238
- /etc/passwd, 227-228, 261, 263-264
- /etc/printcap, 381, 382-395, 397, 551-553
- /etc/rc.d/rc, 172-173
- /etc/rc.d/rc.serial, 417-419
- /etc/resolv.conf, 134-135, 437-438, 456-457, 457-458
- /etc/sendmail/cf, 502
- /etc/services, 441-443, 443-444
- /etc/smb.conf, 546-549
- Ethernet
 - cabluri, 445-447
 - concentratoare, 445-446
 - CSMA/CD (Carrier Sense Multiple Access with Collision Detection) și, 427-428, 444-445
 - descriere generală, 444-453
 - LAN (rețele locale), 137, 426, 444-445, 491-495
 - plăci, 27-28, 76-77, 112-113, 446-450, 452-453, 468-469
 - protocol, 130-131
- eval, comandă, 258
- eveniment(e)
 - acțiuni de la, 274-277
 - conduse prin program, 592-603
 - de mouse, 276-277, 611, 612-613
 - de tastatură, 276-277, 611, 612-613
 - evenimente ButtonPress, 598
 - în fereastră, 277, 611-612, 612-613
 - măști, 598
 - procedura de înregistrare, 597-598
- Excel (Microsoft), 541-542
- EXE (executabile), fișiere
 - extensii de fișier pentru, 30-31
 - FDISK.EXE, 32, 241, 243
 - FIPS.EXE, 36-38
 - LOADLIN.EXE, 40
 - modernizări și, 106-107
 - RAWRITE.EXE, 38, 39
 - RESTORRB.EXE, 37
 - SCSISEL.EXE, 356
 - exec, comandă, 258, 264
 - exec, funcție, 225-227
 - ExecCGI, opțiune, 536-537
 - exit, comandă, 258, 274-275, 477
 - exportul fișierelor, 69-70
 - expr, comandă, 212-213, 258
 - expresii, 215-216, 218-222, 248-249, 252-253
- F**
 - f, opțiune, 266-267
 - factor de formă, 464-465
 - FancyIndexing, opțiune, 534-535
 - FAQ (Frequently Asked Questions), 59, 434, 554-555
 - Fast, plăci Ethernet, 468
 - FAT (tabela de alocare a fișierelor), sistem de fișiere, 32, 113
 - FAT32, sistem de fișiere, 32
 - fc, câmp, 384
 - fdformat, comandă, 239
 - FDISK, comandă, 17, 29-31, 32-36, 48, 241, 243, 314-315, 316-318
 - fdisk, program, 48, 49, 314-315, 316-318
 - FDISK.EXE, 23, 271, 273. *Vezi și* FDISK, comandă
 - fereastră root
 - aspect, 176-179
 - culori pe ecran și, 176-178
 - descriere generală, 175-176
 - proprietăți, 181-183
 - ferestre. *Vezi și* ferestre root (rădăcină)
 - active, 141-142
 - adezive, 185-187
 - descriere generală, 608-609
 - evenimente, 277, 611-612, 612-613
 - identificatori (ID), 622-623
 - înălțime/lățime, 183-184
 - opțiuni referitoare la aspect, 183-184
 - programe de gestionare a, 129-130, 141-142, 165-166, 185
 - ff, câmp, 384
 - FIFO (first-in-first-out), buffere, 399-400
 - figure1.pcx, fișier, 194-195

- figure1.xwd, fișier, 194-195
- File Transfer Protocol (FTP), 435-436, 438-444, 494, 498-499, 505-506, 520-521, 525
 - anonime, 58-59, 499, 500-501, 514-516
 - descriere generală, 429
 - instalare și, 18, 47, 58-59
 - obținerea remediilor cu, 99-100, 101-102
 - pachete, 58-59
 - publicarea informațiilor utilizând, 137
 - TCP/IP și, 131-132
- file, comandă, 258
- FileSelectionBox, primitivă, 603
- filtrarea lucrărilor de tipărit, 392-393
- filtre de intrare (scripturi shell), 386-387, 551-552
- find, comandă, 208-209
- Finlanda, 14-15, 99, 360-361
- FIPS (First Nondestructive Interactive Partition Splitting Program), 29-30, 31, 36-38, 52-53
- FIPS.EXE, 36-38
- fire de execuție, 127-128
- firewall (parașoc), 109-110, 454-455, 493, 544
- firmware, 330
- fișier(e). *Vezi și* nume de fișiere
 - acces la, 227-228
 - căutare, 208-209
 - comprimare, 117, 525-526
 - copiere, 30-31, 233-235, 236, 238-239
 - manipulare, 207-208
 - operații în Tcl, 262-264
 - ștergere, 238
- fișiere de configurare
 - /etc/dosmu.conf, 239-240, 241-242, 244-245
 - /etc/host.conf, 134-135, 437-438, 457-458
 - /etc/httpd/conf/access.conf, 529-530
 - /etc/httpd/conf/httpd.conf, 530-531
 - /etc/httpd/conf/srm.conf, 529-530, 532-536
 - /etc/inetd.conf, 442-444
 - /etc/lilo.conf, 118-119, 122-123, 284, 318-319, 452-453
 - /etc/mtools.conf, 237-238
 - /etc/resolv.conf, 134-135, 437-438, 456-457, 457-458
 - /etc/smb.conf, 546-549
 - fișiere-jurnal
 - instalare, 59-60
 - înregistrarea erorilor, 69-70, 388-389, 531-532
 - Fizic, nivel, 427-428, 428-429
 - flush, comandă, 258, 477
 - fo, câmp, 384
 - focus, comandă, 268-269
 - FollowSymLinks, opțiune, 536-537
 - fonturi, 182-183, 184-185, 663
 - descriere generală, 609
 - desenarea textului și, 621-622
 - for, cicluri, 212-213, 215-216, 225, 253-257
 - for, comandă, 255, 258
 - for, instrucțiuni, 212-213, 215-216, 225, 253-257
 - for, structură de control, 212-213
 - foreach, comandă, 225, 255, 258
 - fork, funcție, 225-227
 - form feed (f), caracter, 250-251, 387-388, 419-420
 - Form, primitivă, 603
 - form.C, 565-566
 - form.h, 565-566
 - form.o, 565-566
 - FORMAT, comandă, 30-31, 31-32, 36-37, 238, 242, 258
 - FORMAT.COM, 32
 - FORTRAN, 127-128, 218-219, 219-220, 385
 - fosfor, 140, 294-295
 - FoxPro, 135
 - FQDN (numele complet al domeniului), 436-438
 - frame, comandă, 267-268
 - Frame, primitivă, 603
 - frecvențe, sincronizare, 62-63, 143-144, 148, 161, 301-302, 303
 - Free Software Foundation, 557
 - fs, câmp, 384
 - fsck, program, 323
 - FTP (File Transfer Protocol), 435-436, 438-444, 494, 498-499, 505-506, 520-521, 525
 - anonime, 58-59, 499, 500-501, 514-516
 - descriere generală, 429
 - instalare și, 18, 47, 58-59

obținerea remediilor cu, 99, 99-100, 101-102
 pachete, 58-59
 publicarea informațiilor utilizând, 137
 TCP/IP și, 131-132
 ftp.funet.fi, sit FTP, 99, 102
 Fujitsu, plăci Ethernet, 27-28, 447-448
~~Funai~~, unități de CD-ROM, 25-26
 funcția cos, 285
 funcția CStext, 705
 funcția delete, 243
 funcția DrawnButton, 705
 funcția each, 243
 funcția exec, 252-253
 funcția fork, 225-227
 funcția getenv, 138
 funcția keys, 243
 funcția Label, 705
 funcția List, 705
 funcția log, 285
 funcția open, 227-228
 funcția PushButton, 705
 funcția pwd, 236
 funcția Sash, 705
 funcția ScrollBar, 705
 funcția Separator, 705
 funcția shift, 250
 funcția sin, 285
 funcția sqrt, 285
 funcția system, 128-129, 225-227
 funcția tan, 285
 funcția TearOffButton, 705
 funcția Text, 705
 funcția ToggleButton, 705
 funcția values, 243
 funcția XmNarmCallback, 597
 funcția XmNdisarmCallback, 597
 funcția XtAddCallback, 597
 funcția XtAddEventHandler, 598
 funcția XtAllCallback, 592-593
 funcția XtAppInitialize, 592-593
 funcția XtAppMainLoop, 592-593
 funcția XtCreateManagedWidget, 595-596, 596-597
 funcția XtDestroyWidget, 598-599
 funcția XtRealizeWidget, 592-593
 funcția XtSetMappedWhenManaged, 598-599

funcția XtVaCreateManagedWidget, 592-593, 596-597
 funcții
 descriere generală, 215-216
 funcția CascadeButton, 601-602
 furnizori de servicii Internet (ISP), 499-500, 505-506, 543-544, 545-546
 America Online, 498, 499
 CompuServe, 413, 498, 499
 conexiuni la, 112, 137
 conturi SLIP de la, 474-476
 descriere generală, 132-133, 553-554
 procurarea adreselor IP, 433
 rețele dial-up și, 470-471, 474-476, 484
 fus orar, specificare, 17, 67-68
 fusul orar local, specificare, 17, 67-68
 Future Domain, 24, 323, 326-327
 fwm, program de gestionare a ferestrelor, 175
 fwm2, program de gestionare a ferestrelor, 57, 165, 185-186, 196-197, 368-370

G

Gawk, 667
 gazde. *Vezi și* nume de calculatoare-gazdă
 atribuire, 78
 pentru situri Web, 531
 transpunere în adresă IP, 69
 gcc (compilator C GNU) 96, 136, 541, 559-564
 gdb, program de depanare, 58-59, 557-558, 571-576
 Genoa, plăci video, 20, 304-305
 get, comandă, 101, 258, 477
 GET, comandă, 522, 522-523
 getenv, funcție, 128-129
 gets, comandă, 262-263, 263-264
 Getting Ready to Install, fază de instalare, 44, 45-48
 getty, 169-170, 417-419, 421-423
 gf, câmp, 384
 ghilimele ("), 208-209, 217, 251-252
 Ghostscript, 57, 667, 686-689
 Ghostview, 375-376, 394-397, 667, 689-690
 GIF (Graphics Interchange Format), imagini, 197. *Vezi și* imagini grafice
 GIMP, 667

GISWaveLAN, plăci Ethernet, 27-28
 glob, comandă, 258, 263-264, 264
 global, comandă, 257, 258
 GMT (Greenwich Mean Time), 67
 GNU
 comanda find, 208-209
 emacs, editor, 17, 58-59, 136, 200-201, 214-215, 541, 557-558, 559-560, 651-660, 667, 668
 gcc, compilator, 93-94, 136, 541, 559-564
 gzip, 99, 667, 671-672
 instrumente, ajutor pentru, 557-559, 654-655
 Library General Public License (LGPL), 576, 578
 licența publică (GPL), 448-449, 557, 559-560, 576-578
 make, utilitar, 133-134, 136, 557-558, 563-564
 Gnuplot, 667, 684-686
 programe de depanare (debugger), 136, 557-558, 559-560, 571-576
 GoldStar, unități de CD-ROM, 25-26, 337-338
 Gopher Server, 58-59
 goto, comandă, 477
 goto, instrucțiune, 225-226
 GPL (licență publică GNU), 448-449, 557, 559-560, 576-578
 gpm, comandă, 145-146, 369-370
 gpm, serviciu, 69
 grab, comandă, 268-269
 Gravis, plăci de sunet, 26-27, 348-349, 350-351
 gray, opțiune, 175-176
 grep, comandă, 93-94, 101-102, 122, 192-193, 199, 219-221, 231-232
 groff, 541, 660-667
 grupuri de discuții, 58-59, 78, 98, 133-134, 499, 505-515
 comp.os.linux.setup, 98
 copm.os.linux.announce, 98, 506-507, 510-511, 511-512, 514-515
 gazde în Internet și, 544
 plasarea mesajelor în, 512-514
 software, instalare, 499-501

grupuri de lucru
 descriere generală, 543
 servere, 542-543, 543
 gs, comandă, 397
 GTEK, plăci seriale, 423
 GUI (interfață grafică cu utilizatorul), 14, 20, 129-131, 139, 141-144, 145-146
 gunzip, comandă, 99, 133-134
 gzip (GNU), 99, 667, 671-672
 gzip, comandă, 99, 667, 671-672

H

hardware. *Vezi și* hardware specific
 liste de verificare, creare, 28-38
 modernizări Linux și, 102, 103-104
 hărți de pixeli, 610, 634-638
 hdimage, 241-242
 HDLC (High-Level Data Link Control), 473-474
 HEAD, comandă, 529
 Headland Technologies, plăci video, 21, 305
 hello.C, 562-563
 „Hello, World!“, program, 214-215, 247-248, 265-267, 562-563
 help, comandă, 477
 Hercules monocrom, plăci video, 20
 Hewlett-Packard, 28, 135, 141-142, 169-170, 280-281, 475-476, 544
 hexazecimală, notație
 pentru culori, 176-177
 pentru plăci de sunet, 352-353
 High Sierra File System, 113-114
 hipertext. *Vezi și* HTML (Hypertext Markup Language)
 descriere generală, 520-521
 HTTP și, 521-524
 history, comandă, 200, 258
 hl, câmp, 384
 HOME, variabilă de mediu, 168-169, 196
 HOSTS.TXT, 435-436
 hot swapping, 466-467
 HOWTO, fișiere, 85-90
 HTML (Hypertext Markup Language), 499-500, 545-546, 640, 660-661
 ancoră, în URL, 520-521
 fișiere, prestabilite, 522, 526

- nume de fișiere, 526
 HTTP (HyperText Transfer Protocol), 429, 440-442, 499-500, 518-519, 520-521, 524, 527-529, 531, 533-534
 httpd (HTTP daemon), 69, 528-529, 531-533
- I**
 i, opțiune, 202, 394-395, 460-461
 I/O (intrare/ieșire)
 adrese de port, 288-289, 289-290, 318-319
 autotestare și, 332
 Ethernet și, 449-452
 modemuri și, 401-403
 mouse și, 365
 plăci de sunet și, 349, 355-356
 redirecție, 193-194
 unități de CD-ROM și, 342-343
 iBCS (Intel Binary Compatibility Specification), modul, 129-130, 541
 IBM (International Business Machines), 19, 22-23, 25-26, 399-400, 447-448, 544
 conectoare, 404-405
 PC-uri (calculatoare personale), 15-16, 19, 141, 312-313, 313-314
 plăci video, 20, 305-306, 309-310
 stații de lucru, 135, 169-170
 tastaturi, 358-359
 ic, câmp, 384
 ical, 672
 ICLEtherTeam, plăci Ethernet, 28, 447-448
 ICMP (Internet Control Message Protocol), 460-461, 489-490
 IconsAreLinks, opțiune, 534-535
 ICP-Vortex, 323
 id, câmpuri, 172-173, 316-317
 IDE (Integrated Drive Electronics), interfață, 98, 468
 descriere generală, 312-313
 dispozitive PCI și, 287-288
 instalare și, 22, 26-27, 49
 modernizare Linux și, 98, 107-108, 108-109
 nume de unități, 49
 partiții și, 232-233, 235
 unități de CD-ROM, 341-342, 345-346, 347-348
 ident, utilitar, 578-579
 IEEE (Institute of Electrical and Electronics Engineers), 126, 127-128, 128-129, 445, 445-446, 564-565
 ieșire. *Vezi și* închidere
 din programul de gestionare a ferestrelor, 165
 din programul de gestiune a ferestrelor, 165
 Disk Druid, 52
 Disk Druid, 52
 sendmail, sesiuni, 504-505
 sesiuni de e-mail, 504-505
 X Window, 83-84
 XWindows, 83-84
 IETF (Internet Engineering Task Force), 433
 if, câmp, 386, 389-390
 if, comandă, 253, 259, 261
 if, instrucțiune, 215-216, 223-224, 253-257
 if, structură de control, 212-213
 if-else, instrucțiuni, 253-254
 if-then-else, instrucțiuni, 215-216
 ifconfig, comandă, 460-461, 483, 489
 image1.xwd, 197-198
 ImageMagick, 667
 imagini grafice. *Vezi și* imagini bitmap; culori
 afișare, 634-638
 aplicații, 57, 667, 679-690
 contexte grafice, 609, 614-624, 636-637, 637-638
 desenare, 614-622
 fișiere, trunchiate, 394-395
 GIF (Graphical Interchange Format), 178-179
 imagini PCX, 178-179, 193-194, 394-395
 în World Wide Web, 519-520
 JPEG (Joint Photographic Experts Group), 178-179
 modele de fundal pentru, 177-178
 pachete, 57, 667, 679-690
 raster, 140, 294-296, 634-636
 sigle, 177-178
 imagini raster, 152, 333-335, 743-744. *Vezi și* linii de raster
 imake, 136

- imap, 667
 imprimante
 configurare, 70-74
 copiere la, 376-377
 derularea tipăririi (spooling), 373-374, 376-378, 380-381, 381-382
 descriere generală, 70-90, 372-398
 instalare și, 18, 57, 58-59, 71-72
 la distanță, 391-393
 locale, 390-391
 nume de dispozitiv, 372-373, 383-384
 setare/configurare, 17, 388-392, 551-553
 stare, verificare, 374-376
 trimiterea lucrărilor de tipărit la, 381
 inc, comandă, 477
 Includes, opțiune, 536-537
 IncludesNoExec, opțiune, 536-537
 incr, comandă, 259
 index.html, 522, 526
 Indexes, opțiune, 536-537
 inetd, server, 69, 442-444
 info, comandă, 259
 info, program, 557-559
 init, comandă, 170-175, 196-197, 236, 417-419, 478
 init, proces, 170-175, 196-197, 236, 417-419, 478
 initrd, linie, 118, 119, 122-123
 inițializare, imagini de pe hard-disc, 242-244
 inn, 667
 innd, serviciu, 69
 instalare, Linux
 cele mai importante faze, 44-45
 descriere generală, 14-90
 discuri suplimentare pentru, 39-40, 45
 încărcare Linux pentru, 40-43
 jurnale, 59-60
 pregătirea calculatorului pentru, 18-40
 procesul, înțelegere, 16-17
 program, interacțiunea cu, 42-43
 selectarea componentelor pentru, 55-60
 timp consumat cu, 44
 xdm și, 170-171
 instalare. *Vezi și* instalare, Linux
 nucleu, 98-99
 Red Hat Package Manager, 94-95, 121-122
 software de e-mail, 499-506
 software pentru grupuri de discuții, 499-501
 Install, buton, 47
 Institute of Electrical and Electronics Engineers (IEEE), 126, 127-128, 128-129, 445, 445-446, 564-565
 instrucțiuni
 descriere generală, 215-216
 instrucțiunea do-while, 215-216
 instrucțiunea goto, 225-226
 instrucțiunea if, 215-216, 223-224, 253-257
 instrucțiunea typedef, 594-596
 instrucțiunea while, 215-216, 223-224, 253-257
 instrucțiuni de control al fluxului, 215-216, 223-226
 instrucțiuni for, 212-213, 215-216, 225, 253-257
 instrucțiuni if-else, 253-254
 instrucțiuni if-then-else, 215-216
 instrucțiuni switch, 253-257
 instrucțiuni de control al fluxului
 programului, 215-216, 223-226
 Integrated Drive Electronics (IDE), interfață, 98, 468
 descriere generală, 312-313
 dispozitive PCI și, 287-288
 instalare și, 22, 26-27, 49
 modernizare Linux și, 98, 107-108, 108-109
 nume de unități, 49
 partiții și, 232-233, 235
 unități de CD-ROM, 341-342, 345-346, 347-348
 Intel, 18, 109, 514. *Vezi și* iBCS (Intel Binary Compatibility Specification), modul
 interconectarea rețelelor, 498
 Internet. *Vezi și* World Wide Web
 calculatoare-gazdă, configurare, 497-517
 conexiuni dedicate, 137
 obținerea informațiilor din, 498-499
 resurse Tcl/Tk pe, 247-248
 rețele private, 433
 servicii și numere de port, 441-443

utilizarea termenului, 498
 InterNIC (Internet Network Information Center), 432-433
 intranet, 545-546
 IP (Internet Protocol). *Vezi și* adrese IP
 administrare firewall (parafo), 58-59
 configurare, 44, 65-67
 descriere generală, 429
 mascare, 492, 493-495
 modernizări Linux și, 91, 109-110
 retransmitere, 91
 rețele dial-up și, 472-473
 tunneling, 109-110
 IPC (Inter Process Communication), 106-107, 603-604
 IPX/Netware Connectivity, 58-59
 IPX/SPX (Inter Packet Exchange/Sequenced Packet Exchange), protocol, 58-59, 110, 130-131, 429, 454-455
 IRC (Internet Relay Chat), 494
 ircii, 667
 IRQ (cereri de întrerupere), 288-292, 318-319, 332, 349, 354, 414
 Ethernet și, 450-452
 modemuri și, 402-403
 mouse și, 365
 ISA (Industry Standard Architecture), magistrală, 14, 281-282
 ISDN (Integrated Services Digital Network), 113, 473
 ISO (International Standards Organization), 127-128, 343-344, 427
 fonturi și, 184-185
 modelul de referință OSI, 131-132
 ISP (Internet Service Provider), 499-500, 505-506, 543-544, 545-546
 America Online, 498, 499
 CompuServe, 413, 498, 499
 conexiuni la, 112, 137
 conturi SLIP de la, 474-476
 descriere generală, 132-133, 553-554
 procurarea adreselor IP, 433
 rețele dial-up și, 470-471, 474-476, 484
 ispell, 667, 673-674
 ITT, plăci video, 305-306
 ITU (International Telecommunications Union), 407

I
 începerea lucrului. *Vezi și* pornire
 cu DOSEMU, 243-244
 cu Linux pentru prima dată, 78-90
 închidere. *Vezi și* shutdown, comandă
 după crearea discurilor de boot (încărcare), 79-81
 salvarea parametrilor plăcilor de sunet la, 69-70
 X Window, 83-84
 întrebări adresate frecvent (FAQ), 59, 434, 554-555
 întreruperea alimentării, 321-322

J
 JED, 668-669
 jocuri, 57, 667
 Joe, editor de text, 668-669
 join, comandă, 259, 260, 261
 JPEG (Joint Photographic Experts Group), imagini, 197. *Vezi și* imagini grafice
 jumpere, 284

K
 k, comandă, 104
 k, opțiune, 483
 kbdrate, utilitar, 358-360
 kerneld, serviciu, 69
 keys, funcție, 218
 keytable, serviciu, 69
 kill, comandă, 163-164, 197-198
 Koblas, David, 130-131
 Korn, shell (interpretor), 191-192
 KTI, plăci Ethernet, 28, 447-448

L
 l, opțiune, 386
 L, opțiune, 549
 label, comandă, 267-268
 LABEL, comandă, 238
 Label, funcție, 601-602
 LAN (Local Area Network), 635-636, 638-648. *Vezi și* rețele locale (LAN)
 Lan Manager, 69-71, 132-133, 542-543, 545-554
 lappend, comandă, 259

laptop, calculatoare, 109-110, 115-116, 289-292, 370-371
 Lasermate, unități de CD-ROM, 25-26
 LaTeX, 541
 lățime de bandă, 554-555
 LBA (Linear Block Address), 321
 LCD (ecran cu cristale lichide), ecrane, 291-292, 294-295
 LCP (Link Control Protocol), 473
 ldd, utilitar, 584-585
 led, diodă electroluminiscentă, 358, 363-364
 legături
 descriere generală, 519-521
 simbolice, 377-378
 legături simbolice, 377-378
 less, comandă, 208, 241
 Lesstif, stil WM, 188-190
 lf, câmp, 385, 387-388
 LGPL (GNU Library General Public License), 576, 578
 lha, pachet, 57
 LILO (Linux Loader), 319-320, 320-321, 355, 452-453
 instalare și, 18, 22, 44, 75-79, 318-319
 încărcare de pe hard-disc cu, 318-319
 MBR și, 314-315
 modernizări Linux și, 117, 118-120, 122-123
 probleme, 329
 reconfigurare, 122-123
 unități de CD-ROM și, 342-343
 limbi străine, 359-361, 407
 limbi străine, 359-361, 407
 Limit, opțiune, 536-537
 limită de domeniu (range), operator, 225
 lindex, comandă, 259
 line-feed (salt la linie nouă), caracter, 194-195, 392-394, 412-413, 419-420, 486-487
 linie de redirectare, simbol (|), 192-193, 221, 228, 380
 linii de raster, 152, 156-157, 334, 335, 342.
 Vezi și imagini raster
 linii, desenare, 618-619, 624-629
 liniuță (-), 256-257, 270, 562-563, 657

liniuță de subliniere (_), 201, 607-608
 linsert, comandă, 259
 Linux Journal, 18, 508-509
 Linux Loader (LILO), 319-320, 320-322, 355, 452-453
 instalare și, 18, 21-22, 44, 75-79, 318-319
 încărcare de pe hard-disc cu, 318-319
 MBR și, 314-315
 modernizări Linux și, 117, 118-120, 122-123
 probleme, 330
 reconfigurare, 122-123
 unități de CD-ROM și, 342-343
 Linux, instalare
 cele mai importante faze, 44-45
 descriere generală, 14-90
 discuri suplimentare pentru, 39-40, 45
 încărcare Linux pentru, 40-43
 jurnale, 59-60
 pregătirea calculatorului pentru, 18-40
 procesul, înțelegere, 16-17
 program, interacțiunea cu, 42-43
 selectarea componentelor pentru, 55-60
 timp consumat cu, 44
 xdm și, 170-171
 list, comandă, 259
 List, funcție, 601-602
 listbox, comandă, 267-268
 literele unităților de disc, 28-29, 48
 llength, comandă, 259
 LMS Philips, unități de CD-ROM, 25-26
 lo, câmp, 385
 loadkeys, 360-361
 LOADLIN.EXE, 40
 Local CD-ROM, opțiune, 47
 local, cuvânt-cheie, 169-170
 Location, bara cu instrumente, 527-528
 lock, fișier, 381-382
 log, funcție, 253
 login grafic, configurare, 169-175
 login. *Vezi și* pornire
 configurare, 169-175
 nume de, 196
 pornire X Window System la, 168-169
 Logitech, mouse, 22-23, 145-146, 146-147, 364-365, 367-368

Logitech, plăci de sunet, 26-27, 348-349
 LOGNAME, variabilă de mediu, 196
 Longshine, plăci seriale, 423
 Longshine, unități de CD-ROM, 25-26
 lookup, fișier, 222
 loopback, comunicații, 432-433, 451-452
 Low Voltage Operation, 466-467
 lower, comandă, 268-269
 lp, câmp, 385, 386, 386-387
 lpc status, comandă, 376-377
 lpc, comandă, 377-378, 378-379
 lpd, comandă, 377-378, 380, 381-382, 386-387, 387-388, 390, 392-393
 lpd, serviciu, 69
 lpq, comandă, 374-375, 376-377, 377-378
 lpr, comandă, 376-377, 377-378, 380-381, 386, 391-392, 394-395
 lprm, comandă, 376-377, 377-378
 lps status, comandă, 374-376
 lps, comandă, 373-375
 LPT1 (port paralel), 71-72, 372-373, 376-377, 450
 LPT2 (port paralel), 71-72, 372-373
 lrange, comandă, 259
 lreplace, comandă, 259
 ls, comandă, 191-192, 202, 205-207, 236, 264, 339-340
 lsearch, comandă, 259
 lsort, comandă, 259
 LUN (Logical Unit Numbers), 330
 lungimea cuvântului, 400-401

M

m, comandă, 316-317
 m, opțiune, 315-316
 Macintosh, 178-179
 macro-uri, 607-608, 665-666
 magistrale
 descriere generală, 19
 magistrala AT, 19
 magistrala ISA (Industry Standard Architecture), 14, 281-282
 magistrala MCA (Micro Channel Architecture), 19, 282
 magistrala PCI (Peripheral Component Interconnect), 14, 19, 104-105, 282-283, 287-289, 313-314, 322

magistrala VLB, 19, 282
 tipuri de, rezumat, 279-293
 mail, comandă, 502-503
 mail, utilitar, 128-129
 MainWindow, primitivă, 603
 make boot, comandă, 117
 make clean, comandă, 117
 make config, comandă, 104, 105-106, 493
 make dep, comandă, 117
 make xconfig, comandă, 104
 make, utilitar (GNU), 133-134, 136, 557-558, 563-564
 makefile, fișiere, 564-566, 567-570, 580-581, 583, 593-595
 man bash, comandă, 212-213
 man, comandă, 662
 man, pagini, 59, 84-85, 660-666
 Manager, 542-543
 Manager, 542-543
 Manager Widgets, 598-599, 603-604
 adrese IP și, 431-432
 client/server, 141
 conectare, la Internet, 137
 descriere generală, 426
 Ethernet, 137, 426, 444-445, 491-495
 instalare și, 65-67, 69-71
 modernizări Linux și, 91, 109-110
 partajarea fișierelor, 499-500
 rutare și, 491-495
 maplay, pachet, 58
 margini
 culoare, 182-185
 lățime, 182-185
 MARS, 69
 mars-new, serviciu, 69
 masive
 asociative, 216-217, 218-219
 descriere generală, 217-218
 operatori indici, 218-219
 Tcl/Tk, 262
 Matrox, plăci video, 20, 305, 310
 Matsushita, unități de CD-ROM, 25-26, 337-338
 mattrib, comandă, 238
 MBR (Master Boot Record), 75, 243-244, 314-315, 317-322

MCA (Micro Channel Architecture),
 magistrală, 19, 282
 MCI, 553-554
 mcopy, comandă, 236, 238
 MD, comandă, 238
 mdel, comandă, 238
 mdir, comandă, 236, 238, 238-239
 Media Player, 355
 Media TriX, plăci de sunet, 26-27, 348-349
 Media Vision, 24, 26-27, 322-323, 348-349
 memorie video, 152, 172-173. *Vezi și* VRAM (video RAM)
 memorie virtuală, 17, 29-30
 memorie. *Vezi și* nucleu; RAM (memorie cu acces aleatoriu)
 meniuri, cerințe pentru stiluri specifice, 141-142
 menu, comandă, 267-268
 menubutton, comandă, 267-268
 merg, utilitar, 578-579
 mesaje
 procesul de boot (încărcare), 41-43
 vizualizare, 41-43
 message, comandă, 267-268
 MessageBox, primitivă, 603
 Metcalfe, Robert M., 445
 Metro Link, 309-310
 Metro-X, 309-310
 mformat, comandă, 238, 239
 microfoane, 348-349
 microprocesoare
 convenții de nume pentru, 286-287
 descriere generală, 279-293
 instalare și, 14-15, 19, 44, 78-79
 interpretorul Bash și, 196-198
 modernizări Linux și, 93, 104-105, 107-108
 MS-DOS și, 239-240
 tipuri de, 279-293
 Microsoft Access, 541-542
 Microsoft Excel, 541-542
 Microsoft Office, 541-542
 Microsoft Sound System, 26-27, 291-292, 348-349, 350-351
 Microsoft Windows 3.x, 14-15, 18, 28-31
 Microsoft Windows 95
 bara de operații, 187

 cursoare și, 178-179
 FDISK și, 32
 folosit în mediu de afaceri, 540, 545-546, 549
 instalare și, 18, 21-22, 28-31, 32, 40-41, 70-71, 82
 în rețele dial-up, 494-495
 LILO și, 317-318
 partiționare hard-disc sub, 28-31
 suprafața de lucru, 185-190
 Microsoft Windows NT
 folosit în mediu de afaceri, 541, 553-554
 instalare și, 18, 70-71
 standarde POSIX și, 127-128
 Microsoft Word, 541-542, 545
 Microsoft, mouse, 22-23, 145-146, 364-365
 MicroSolutions, unități de CD-ROM, 25-26
 MIDI, plăci de sunet, 26-27, 348-349, 350-351, 356
 Midnight Commander, 667, 674-675
 MIME (Multipurpose Internet Mail Extensions), 532-535
 minfree, 387-388
 minibuffere, 652-653
 minicom, 414-418, 471
 Minix, sisteme de fișiere, 113
 MIPS, procesoare, 280-281
 MIT (Massachusetts Institute of Technology), 141-142, 143
 Mitsumi, unități de CD-ROM, 25-26, 337-338, 347-348
 mixaj, programe de, 355
 mkdir, comandă, 129
 MKDIR, comandă, 238
 mke2fs, comandă, 323
 mkinitrd, comandă, 120, 122
 mlabel, comandă, 238
 mmd, comandă, 238
 modeline, 302-304, 652-653
 modem(uri)
 apeluri externe prin, 413-414
 cabluri/conectoare, 405-407, 413-414, 421-423
 comenzi, 408-413
 descriere generală, 395-397
 instalare, 413-414
 PC Card, 468

- rețele dial-up, 487
- standarde, 407-408
- viteze, 137, 525
- modem, comandă, 478
- modem, opțiune, 487
- modernizări (upgrade), 91-124, 133-134
- modificator, taste, 358-359
- modul comandă vizuală, 647
- module încărcabile, 129
- module, nucleu, 103-104, 117
- moduri video, 140, 162
- monitor
 - descriere generală, 20-22, 294-311
 - imagini bitmap și, 140
 - importantă, 297-298
 - instalare și, 19-22, 62-64
- monitorizare
 - performanțe sistem, 134-135
 - rețele, 134-135
 - parametri pentru lățimea de bandă, 144-145
 - rezoluții, 158, 164-165, 295-296
 - tipuri de, selectare, 62-64
 - viteze de redesenare, 140
 - X Window System și, 140, 143-145, 158-159, 160-162, 163-164
- montare, 145, 259-265. *Vezi și* puncte de montare
- more, comandă, 208
- Motif (X Window System), 129-131, 246, 264, 526-527
 - descriere generală, 139
 - mouse și, 369-370
 - pachet (toolkit), 591-592
 - programare, 591-603
 - stil, 188-190
- mount, comandă, 98, 232-233, 233-234, 235, 239-240
- mouse
 - alternative la, 369-371
 - configurare, 60-61
 - cu trei butoane, 367-368
 - descriere generală, 364-371
 - evenimente, 276-277, 611, 612-613
 - instalare și, 22-23, 28-29, 60-61
 - interfețe, 364-365
 - modernizări Linux și, 114-115
 - nume de dispozitiv, 366, 367
 - protocoale, 366-368
 - tipuri acceptate de Linux, 22-23
 - X Window System și, 145-147, 160-161
- mouseconfig, 60-61
- Moxa, plăci seriale, 423
- MPU-401 MIDI, plăci de sunet, 26-27
- mrd, comandă, 238
- mread, comandă, 238-239
- mren, comandă, 238-239
- MS-DOS
 - „secrete“, 231-245
 - BACKUP, utilitar, 31
 - comenzi, 30-31
 - conexiune la, întreținere, 135
 - emulator (DOSEMU), 57, 135, 239-245, 667, 669-670
 - fișiere de comenzi (batch), 195
 - fișiere, montare, 134-135
 - instalare și, 14-16, 22-23, 28-41, 52-54, 77
 - interpretorul Bash și, 191-192, 194-195, 197-199, 202-203
 - LILO și, 317-318
 - lucrul în, după instalarea Linux, 29-31
 - mod, repornire în, 38, 41
 - modemuri și, 399-400, 402-404, 421-422
 - modernizări Linux și, 105-106, 113-114, 118
 - nume de fișiere, 30-31, 48
 - partiții, 28-31, 52-54, 77, 231-236
 - plăci de sunet și, 355
 - procesoare și, 279
 - programe de shell (interpretor) și, 194-195
 - repartiționare sub, 31-36, 37-38
 - sisteme de fișiere, 231-236
 - structura de directoare, 202-203
 - tastaturi și, 357
 - tipărire și, 376-377, 392-393
 - TSR (programe rezidente), 105-106, 129
- mttools, utilitare, 57, 135, 236-240, 315-316
- mtype, comandă, 238-239
- multi, opțiune, 457-458
- Multifunction Capability, 466-467
- multimedia, 24-26, 57, 520-521
- multiport, plăci seriale, 421-424
- mutare
 - blocuri de text, 659-660

- cursoare, 43, 45
 - directoare, 117
 - mv, comandă, 207
 - mwm (Motif Window Manager), 130-131
 - mwwrite, comandă, 238-239
 - mx, câmp, 385, 386, 387-388
- N**
- named, serviciu, 69
 - National Semiconductor, 399-401, 448-449
 - Navigation, bară cu instrumente de navigare, 527-528
 - NCP (Network Control Protocol), 473
 - NCR, 24, 305, 323, 326
 - NCSA (National Center for Supercomputing Applications), 526, 528-529
 - NeoMagic, plăci video, 310
 - NetBEUI (Network BIOS Extended User Interface), 429
 - NetBIOS, 429
 - netcfg, script, 456-460
 - netconfig, 459-460
 - Netcraft Web Server Survey, 529
 - netmask, comandă, 478
 - Netscape, browser de Web, 613-617, 631, 783. *Vezi și* browsere de Web
 - netstat, comandă, 461-462
 - NetWare (Novell), 132-133, 429, 539
 - NETWORKING, variabilă, 459-460
 - New Media, plăci Ethernet, 28, 447-448
 - .newsr, fișier, 506-510, 512
 - next, cuvânt-cheie, 309-310
 - NextStep, stil de organizare a suprafeței de lucru, 188-190
 - nf, câmp, 385
 - NFS (Network File System), 113-114, 441-442
 - descriere generală, 132-133
 - instalare și, 18, 47, 52, 58-59, 69-70
 - nfsfs, serviciu, 69-70
 - NIC (Network Information Center), 435-436
 - NIS (National Institute of Standards and Technology), 126
 - niveluri
 - nivelul Aplicație, 427-428, 428-429
 - nivelul Fizic, 427-428, 428-429
 - nivelul Legătură de date, 427-428, 428-429
 - nivelul Prezentare, 427-428
 - nivelul Rețea, 427-428, 428-429
 - nivelul Sesiune, 427-428
 - nivelul Transport, 427-428, 428-429
 - niveluri de rulare, 170-172, 172-175
 - niveluri de rulare, câmp, 172-173
 - NNTP (network News Transfer Protocol), 441-442, 505-506, 509-510, 520-521
 - NNTPSERVER, variabilă de mediu, 506-507, 507-508, 509
 - noduri, 137
 - None, opțiune, 535-536, 536-537
 - Norton, utilitare
 - Disk Doctor, 37
 - Speed Disk, 37
 - NoTrapSignals, indicator, 160-161
 - Novell, plăci Ethernet, 28, 447-448
 - NO_LOCK, 284
 - nucleu
 - adăugarea suportului pentru rețea la, 134-135
 - compilare, 93-94
 - daemon, 105-106
 - descriere generală, 97, 541
 - deteriorat, 79-80
 - instalare și, 22, 40, 41, 69, 75, 79-80
 - încărcare, 40, 41, 118-120
 - încercarea unui nou, 123
 - modernizări și, 96-122
 - module, 103-104, 117
 - mouse și, 365
 - plăci de sunet și, 353-355
 - probleme, 332
 - /proc, sistem de fișiere și, 284-290
 - profilare, 1169-117
 - regenerare, 91-92, 102-120
 - RPM, 120-122
 - rularea unui nou, 119-120
 - suport pentru Ethernet, 449-450
 - sursă, instalare, 98-99
 - TCP/IP și, 131-132
 - unități de CD-ROM și, 339-340
 - null, caracter, 486-487
 - Number Nine Imagine, plăci video, 20, 283, 310
 - nume de căi
 - absolute, 205

descriere generală, 159-160
 nume de domenii, 64, 608-609. *Vezi și* DNS (Domain Name Service)
 nume de fișiere
 extensii, 533-534, 534-535
 HTML, 526
 în URL-uri, 520-521
 MS-DOS, 30-31
 prestabilite, 526
 numele elementelor, 262
 numere de unitate logică (LUN), 330
 numerice, câmpuri, 383-384
 Nvidia, 20, 305

o, opțiune, 562
 OAK, plăci de sunet, 26-27, 348-349
 OAK, plăci video, 305-307
 obținere de informații, din Internet, 498-499
 of, câmp, 385
 Office (Microsoft), 541-542
 OK, buton, 52, 59
 OLIT (OPEN LOOK Intrinsic Toolkit), 591
 ONBOOT, variabilă, 459-460
 online, mod pentru modemuri, 408
 OPEN LOOK, 129-130, 141, 369-370, 591
 Open, casetă de dialog, 244-245
 open, comandă, 259, 262-263
 open, funcție, 227-228
 operatori
 descriere generală, 218-220
 operator cmp, 218-219
 operator de delimitare a unui domeniu, 225
 operator de repetiție, 219-220
 operator eq, 218-219
 operatori aritmetici, 218-219
 operatori de comparație, 218-219
 operatori de indexare într-un masiv, 218-219
 operatori de listă vidă, 219-220
 operatori logici, 253
 operatori aritmetici, 218-219
 operatori de comparare, 218-219
 operatori de listă vidă, 219-220
 operatori logici, 253
 OPTi, plăci de sunet, 26-27

Optics Storage Dolphin, unități de CD-ROM, 25-26, 337-338
 opțiuni (listate după nume)
 opțiunea /S, 32
 opțiunea Add/Remove Programs, 30-31
 opțiunea All, 535-536, 536-537
 opțiunea AuthConfig, 536-537
 opțiunea bitmap, 175-176, 177-178
 opțiunea c, 582
 opțiunea Check for bad blocks during format, 55
 opțiunea CONFIG_APM, 115-116
 opțiunea CONFIG_ATIXL_BUSMOUSE, 115-116
 opțiunea CONFIG_BLK_DEV_FD, 107-108
 opțiunea CONFIG_BLK_DEV_HD_IDE, 107-108
 opțiunea CONFIG_BLK_DEV_IDECD, 107-108
 opțiunea CONFIG_BLK_DEV_IDEFLOPPY, 107-108
 opțiunea CONFIG_BLK_DEV_IDESCSI, 108-109
 opțiunea CONFIG_BLK_DEV_IDETAPE, 107-108
 opțiunea CONFIG_BLK_DEV_IDE_PCMCIA, 108-109
 opțiunea CONFIG_BLK_DEV_LOOK, 108-109
 opțiunea CONFIG_BLK_DEV_LRAM, 108-109
 opțiunea CONFIG_BLK_DEV_MD, 108-109
 opțiunea CONFIG_BLK_DEV_SD, 110
 opțiunea CONFIG_BLK_DEV_TRITON, 108-109
 opțiunea CONFIG_BUSMOUSE, 115-116
 opțiunea CONFIG_EXT2_FS, 113
 opțiunea CONFIG_EX_FS, 113
 opțiunea CONFIG_FAT_FS, 113
 opțiunea CONFIG_FIREWALL, 109-110
 opțiunea CONFIG_FTAPE, 115-116
 opțiunea CONFIG_HPRS_FS, 113-114

opțiunea CONFIG_IDE_CHIPSETS, 108-109
 opțiunea CONFIG_INET, 109-110
 opțiunea CONFIG_IPX, 110
 opțiunea CONFIG_IP_ACCT, 109-110
 opțiunea CONFIG_IP_FORWARD, 109-110
 opțiunea CONFIG_IP_MULTICAST, 109-110
 opțiunea CONFIG_IP_ROUTER, 109-110
 opțiunea CONFIG_ISO9660_FS, 113-114
 opțiunea CONFIG_MATH_EMULATION, 104-105, 106-107
 opțiunea CONFIG_MAX_16M, 106-107
 opțiunea CONFIG_MINIX_FS, 113
 opțiunea CONFIG_MS-DOS_FS, 113-114
 opțiunea CONFIG_MS_BUSMOUSE, 115-116
 opțiunea CONFIG_NET, 106-107
 opțiunea CONFIG_NET_IPIP, 109-110
 opțiunea CONFIG_NFS_FS, 113-114
 opțiunea CONFIG_PCI, 104-105
 opțiunea CONFIG_PRINTER, 114-115
 opțiunea CONFIG_PROC_FS, 113-114
 opțiunea CONFIG_PSMOUSE, 115-116
 opțiunea CONFIG_QICO2_TAPE, 115-116
 opțiunea CONFIG_QUOTA, 113
 opțiunea CONFIG_ROOT_FS, 113-114
 opțiunea CONFIG_RTC, 115-116
 opțiunea CONFIG SCSI, 110
 opțiunea CONFIG SCSI_GDTH, 111
 opțiunea CONFIG_SMB_FS, 113-114
 opțiunea CONFIG_SMB_WIN95, 113-114
 opțiunea CONFIG_SYN_COOKIES, 109-110
 opțiunea CONFIG_SYSVIPC, 106-107
 opțiunea CONFIG_SYSV_FS, 113-114
 opțiunea CONFIG_UFS_FS, 113-114
 opțiunea CONFIG_UMISC, 115-116
 opțiunea CONFIG_UMSDOS_FS, 113-114
 opțiunea CONFIG_VFAT_FS, 113-114
 opțiunea CONFIG_WATCHDOG, 115-116
 opțiunea CONFIG_XIA_FS, 113

opțiunea connect, 487
 opțiunea connect, 487
 opțiunea connection, 488
 opțiunea crtsets, 487
 opțiunea cursor_name, 178-179
 opțiunea d, 194-195
 opțiunea defaultroute, 487
 opțiunea display, 183-184
 opțiunea e, 214-215
 opțiunea ExecCGI, 536-537
 opțiunea f, 266-267
 opțiunea FancyIndexing, 534-535
 opțiunea FollowSymLinks, 536-537
 opțiunea gray, 175-176
 opțiunea i, 202, 394-395, 460-461
 opțiunea IconsAreLinks, 534-535
 opțiunea Includes, 536-537
 opțiunea IncludesNoExec, 536-537
 opțiunea Indexes, 536-537
 opțiunea k, 483
 opțiunea l, 386
 opțiunea L, 549
 opțiunea Limit, 536-537
 opțiunea Local CDROM, 47
 opțiunea m, 315-316
 opțiunea modem, 487
 opțiunea multi, 457-458
 opțiunea None, 535-536, 536-537
 opțiunea o, 562
 opțiunea order, 457-458
 opțiunea p, 641-642
 opțiunea P, 373-374
 opțiunea print, 208-209
 opțiunea r, 90
 opțiunea s, 359-360
 opțiunea ScanHTMLTitles, 534-535
 opțiunea solid, 175-176
 opțiunea Stick/Unstick, 185-187
 opțiunea SuppressDescription, 534-535
 opțiunea SuppressLastModified, 534-535
 opțiunea SuppressSize, 534-535
 opțiunea SymLinksIfOwnerMatch, 536-537
 opțiunea t, 462, 476, 480-481
 opțiunea u, 582
 opțiunea WM Style, 188-189
 opțiunea x, 196-197

Orchid, 20, 306-307, 348-349
 order, opțiune, 457-458
 orizontală, sincronizarea frecvențelor, 62-63,
 143-144, 148, 161, 301-302, 303
 OS/2, 17, 75, 113-114, 141, 317-318, 540
 OSI (Open Systems Interconnection), model,
 131-132, 427-428
 nivelul Aplicație, 427-428, 428-429
 nivelul Fizic, 427-428, 428-429
 nivelul Legătură de date, 427-428, 428-429
 nivelul Prezentare, 427-428
 nivelul Rețea, 427-428, 428-429
 nivelul Sesiune, 427-428
 nivelul Transport, 427-428, 428-429
 Ousterhout, John, 246, 247-248
 OverrideShell, primitivă, 599-601

P

p, comandă, 316-317
 p, opțiune, 373-374
 p2c, 667
 pachet interfață radio, 112
 pachete
 descriere generală, 56-60
 grafice, 57, 667, 679-690
 modernizări Linux și, 93-96
 pachetul apache, 95-96, 96
 pachetul aumix, 58
 pachetul cdp, 58
 pachetul elm, 57, 500-501, 503-504, 667
 pachetul lha, 57
 pachetul maplay, 58
 pachetul pine, 57, 456, 667
 pachetul playmidi, 58
 pachetul portmap, 58-59
 pachetul postgresql, 58-59
 pachetul postgresql-data, 58-59
 pachetul postgresql-devel, 58-59
 pachetul rhsound, 58
 pachetul samba, 58-59
 pachetul smbfs, 58-59
 pachetul sndconfig, 58
 pachetul tm, 57, 667
 pachetul unzip, 57
 pachetul xbill, 57
 pachetul xboard, 57

pachetul xboing, 57
 pachetul xchomp, 57
 pachetul xdemineur, 57
 pachetul xevil, 57
 pachetul xfishtank, 57
 pachetul xgalaga, 57
 pachetul xgammon, 57
 pachetul xgopher, 57
 pachetul xjewel, 57
 pachetul xlander, 57
 pachetul xpat2, 57
 pachetul xpilot, 57
 pachetul xpuzzles, 57
 pachetul xsnow, 57
 pachetul xtrojka, 57
 pachetul zip, 57
 pack, comandă, 268-269, 270-272
 paginare, 51
 pagini banner, 387-388
 pagini Web, 518, 518-519, 526-527, 545-546
 Panasonic, unități de CD-ROM, 25-26,
 337-338
 PanedWindow, primitivă, 603-604
 parametri, comunicații, 400-402
 paranteze pătrate ([]), 56, 69, 105-106,
 198-199, 220-221
 PARC (Xerox Palo Alto Research Center),
 141, 445
 PARITY, biți, 401-402
 parity, comandă, 478
 parole, 17, 83-84
 de root, 74, 78-80
 pornire și, 170-171, 174
 recuperare, 78-80
 schimbare, 78-79
 parole de root, 74, 78-80
 partiții. *Vezi și* FDISK, comandă
 active, 35-36
 atribute, modificare, 52
 configurare, 52-54
 creare, 34-36, 44, 48-60
 descriere generală, 17, 28-32, 48-60,
 314-315
 dimensiuni, 37-38
 DOS, 28-31, 52-54, 77, 231-236
 extinse, 34-35

formatare, 55
 informații, restaurare, 52
 informații, verificare, 33-34
 LILO și, 75-77
 nume, 49
 primare, 34-35, 35-36
 reparaționare, 29-30, 30-31, 32-38
 restaurare, 36
 swap (de schimb), 17, 48, 50-51
 ștergere, 34-35, 52, 52-53
 tabele, actualizare, 52
 tipuri, recunoscute de Linux, 316-318
 PASC (Portable Application Standards
 Committee), 128-129
 patch, comandă, 101, 667, 675-676
 PATH, variabilă de mediu, 146-147, 193-194,
 195-196, 211-212, 213-214, 218, 247,
 262-263, 265
 pauze, 478, 484-485
 PBM (Portable Bitmap), set de instrumente,
 193-194
 PC Cards. *Vezi și* PCMCIA, dispozitive
 conectoare pentru, 464-465
 descriere generală, 464-469
 instalare și, 45
 modernizări Linux și, 108-109
 specificații fizice pentru, 464-466
 pc, câmp, 385
 PCI (Peripheral Component Interconnect),
 magistrală, 14, 19, 104-105, 106-107,
 282-283, 287-289, 313-314, 322
 PCL (Printer Control Language), 381-382
 PCMCIA Card Services for Linux (Card
 Services), 464, 466-468
 PCMCIA, dispozitive, 31-32, 38, 537-544.
Vezi și PC Cards
 PCX, imagini, 197, 214, 453. *Vezi și* imagini
 grafice
 PDF (Portable Document Format), 407
 Pelix, 421-422
 Perl, 191, 667
 descriere generală, 213-214, 215-216
 funcții înglobate, 230
 instalare și, 17, 56, 57-58, 213-214
 script, primul dumneavoastră, 214-216
 sintaxa, 215-217
 permisiuni, pentru directoare, 205-207
 Personal Sound System (PSS), placă de sunet,
 348-349, 350-351
 Personal, bara cu instrumente, 527-528
 pf, câmp, 385
 Philips, unități de CD-ROM, 337-338
 PID (identificator de proces), 196-197,
 444-445, 531-532
 pid, comandă, 259
 pine, pachet, 57, 456, 667
 ping, comandă, 489-490
 pixeli
 biți pe (bpp), 159, 162, 299, 307-309, 310
 descriere generală, 140, 294-295
 place, comandă, 268-269, 270-271, 273-274,
 274-275
 playmidi, pachet, 58
 plăci de sunet
 acceptate de Linux, 26-27, 348-349
 descriere generală, 26-27, 268-269,
 347-356
 drivere, 349-351
 instalare și, 18, 25-27, 69-70
 modernizări Linux și, 102, 115-117
 nume de dispozitiv, 350-352
 parametri, salvare/restaurare, 69-70
 rezolvarea problemelor, 353-356
 testare, 352-353
 unități de CD-ROM și, 25-26
 plăci pentru rețele fără cablu, 468
 plăci video
 accelerate, 305-309
 comune, 304-305
 descriere generală, 20-22, 140, 294-309
 imagini bitmap și, 140
 importanța, 297-298
 instalare și, 15-16, 18-22, 28-29, 62, 64-66,
 80-81
 X Window System și, 129-131, 140,
 143-146, 149, 157-165, 283
 Plug and Play, 466-467
 poartă de acces (gateway), 66-67, 434-435,
 492
 Point-to-Point Protocol (PPP), 28, 426, 471,
 553-554
 clienți, conectarea la rețele de la distanță,
 ca, 562-571
 configurarea calculatorului-gazdă și, 497

- descriere generală, 131-134, 451-452, 473-474
- modemuri și, 413
- modernizări Linux și, 112
- rețele dial-up și, 470-471
- routing, 491-495
- servere, configurare, 494-495
- poligoane, desenare, 619-621
- pornire. *Vezi și* login; începerea lucrului cu
 - personalizare, 168-190
 - renunțare, 163-164
 - restaurarea parametrilor plăcii de sunet la, 69-70
- port, comandă, 478
- portmap, pachet, 58-59
- portmap, serviciu, 69-70
- porturi paralele, 372-373
 - LPT1, 71-72, 372-373, 376-377, 450
 - LPT2, 71-72, 372-373
- porturi seriale
 - COM1, 365, 372-373, 402-404, 414, 421
 - COM2, 365, 372-373, 402-404
 - COM3, 402-404
 - COM4, 402-404
 - imprimante și, 372, 372-373
 - modemuri și, 399, 399-403, 414, 417-419, 421-423
 - modernizări Linux și, 114-115
 - mouse și, 364, 367-368, 369-370
- porturi, paralele
 - LPT1, 71-72, 372-373, 376-377, 450
 - LPT2, 71-72, 372-373
- porturi, seriale
 - COM1, 365, 372-373, 402-404, 414, 421
 - COM2, 365, 372-373, 402-404
 - COM3, 402-404
- porturi. *Vezi și* porturi, paralele; porturi, seriale
 - adrese, 520-521
 - numere, 441-443
 - servere Web și, 520-521, 531
- POSIX (Portable Operating System Interface), standarde
 - descriere generală, 126, 127-128
 - standardul POSIX.1, 127-128, 129, 136
 - standardul POSIX.1a, 127-128
- POST, comandă, 522-523
- Postgres (SQL) Server, 58-59
- postgresql, pachet, 58-59
- postgresql-data, pachet, 58-59
- postgresql-devel, pachet, 58-59
- PostScript, limbaj de imprimantă, 375-376, 381-382, 394-397
- ppl, comandă, 480-481
- ppmtpcs, program, 193-194
- PPP (Point-to-Point Protocol), 28, 426, 471, 553-554
 - clienți, conectarea la rețele de la distanță, ca, 483-491
 - configurarea calculatorului-gazdă, 497
 - descriere generală, 131-134, 451-452, 473-474
 - modemuri și, 413
 - modernizări Linux și, 112
 - rețele dial-up, 470-471
 - routing, 491-495
 - servere, configurare, 494-495
- pppd, 483, 484-485, 486-492, 667
- Presentation Manager, 141
- Prezentare, nivel, 427-428
- primitive (widget) autonome, 598-599, 601-603
- primitive shell (shell widgets), 598-599, 599-601
- primitive
 - desenare în fereastra primitivei, 598
 - primitiva ArrowButton, 601-602
 - primitiva Command, 603
 - primitiva Core, 598-600
 - primitiva DrawingArea, 603
 - primitiva FileSelectionBox, 603
 - primitiva Form, 603
 - primitiva Frame, 603
 - primitiva MainWindow, 603
 - primitiva MessageBox, 603
 - primitiva OverrideShell, 599-601
 - primitiva PanedWindow, 603-604
 - primitiva PushButton, 595-596, 597
 - primitiva RowColumn, 603-604
 - primitiva Scale, 603-604
 - primitiva ScrolledWindow, 603-604
 - primitiva SelectionBox, 603-604
 - primitiva TopLevelShell, 599-601

- primitiva XmDrawingArea, 597-598
- primitive (widget) shell, 598-599, 599-601
- print, comandă, 228
- PRINTER, variabilă de mediu, 380
- Pro Audio, 328, 348-349, 355-356
- probeonly, mod, 158
- problema anului 2000, 22-23
- proc, comandă, 257, 259
- proceduri, 248-249, 257
- proceduri de gestionare a geometriei, 270-274
- proces, câmp, 172-173
- procese, 169-170
- procesoare
 - convenții de nume pentru, 286-287
 - descriere generală, 279-293
 - instalare și, 14-15, 19, 44, 78-79
 - interpretorul Shell și, 196-198
 - modernizări Linux și, 93, 104-105, 107-108
 - MS-DOS și, 239-240
 - tipuri de, 279-293
- procesoare în virgulă mobilă (coprocesoare matematice), 104-105, 106-107, 280
- Procomm, 399-400, 421-422
- Procomm Plus, 421-422
- programe rezidente (TSR), 105-106, 129
- programe shell (interpretor), 138, 214-216.
 - Vezi și* scripturi
 - descriere generală, 208-210
 - în Bash, 208-213
 - în GNU Emacs, 659-660
 - simple, 209-211
- Properties, etichetă, 37
- protejat, mod, 280
- protocoale (listate după nume). *Vezi și* File Transfer Protocol (FTP); IP (Internet Protocol); Point-to-Point Protocol (PPP); Serial Line Internet protocol (SLIP); TCP/IP (Transmission Control Protocol/Internet Protocol)
 - cu conexiune, 439, 440
 - dip (Dial-up Protocol Driver), program, 475-483, 484, 667
 - fără conexiune, 439-440
- HTTP (HyperText Transfer Protocol), 429, 440-442, 499-500, 518-519, 520-521, 524, 527-529, 533-534
- ICMP (Internet Control Message Protocol), 460-461, 489-490
- LCP (Link Control Protocol), 473
- NCP (Network Control Protocol), 473
- NNTP (Network News Transfer protocol), 441-442, 505-506, 509-510, 520-521
- protocolul AppleTalk, 110
- protocolul AX.25 packet radio, 110
- protocolul BOOTP, 476
- protocolul IPX/SPX (Internet Packet Exchange/Sequenced Packet Exchange), 58-59, 110, 130-131, 429, 454-455
- protocolul token-ring, 130-131
- protocolul X, 139, 141
- RIP (Routing Information Protocol), 69-70
- SMTP (Simple Mail Transfer Protocol), 429, 501-502, 504-506
- SNMP (Simple Network Management Protocol), 58-59, 69-70, 429, 439-440
- TFTP (Trivial File Transfer Protocol), 441-442
- UDP (User Datagram Protocol), 429, 439-440
- protocoale cu conexiune, 439, 440
- protocoale fără conexiune, 439-440
- protocoale. *Vezi și* protocoale (listate după nume)
 - descriere generală, 130-131, 428-429
 - în URL-uri, 520-521
- proxyarp, comandă, 478
- ps, comandă, 196-198, 390, 443-444
- PS/2, mouse, 22-23, 60, 145-146, 146-147, 364-365, 370-371
- PS1, variabilă de mediu, 196
- pseudonime, 201-202
- psh.pl, 226-227
- pstgresql, serviciu, 69-70
- punct și virgulă (;), 216-217, 249-250, 251-252, 547-548
- punct(.), 180, 181-182, 205, 207, 210-211, 269-270, 534-535, 641-642, 642-643
- puncte de întrerupere, 571-572
- puncte de montare, 42, 44, 46, 48, 360. *Vezi și* montare

puncte de suspensie (...), 199, 216-217
 PureData, plăci Ethernet, 28, 447-448
 PushButton, funcție, 601-602
 PushButton, primitivă, 595-596, 597
 puts, comandă, 259, 262-263
 pw, câmp, 385
 pwd, comandă, 191-192, 204-205, 259
 pwd, funcție, 212-213
 px, câmp, 385
 py, câmp, 385
 Python, 667

Q

Qlogic, 323
 Quantum, 24, 324
 QuickPort, mouse, 22-23
 QuickStart, fișier, 241
 quiet, mod, 409-410
 quit, comandă, 478
 QUIT, comandă, 504-505

R

r, opțiune, 90
 Racal-Interlan, plăci Ethernet, 28, 447-448
 radiobutton, comandă, 267-268
 raise, comandă, 268-269
 RAM (memorie cu acces aleatoriu). *Vezi și* memorie
 discuri, 118, 122
 instalare și, 20, 50-51
 modernizări Linux și, 108-109, 118, 122
 partiții swap și, 50-51
 plăci de memorie, 468
 procesoare și, 280
 RAMDAC, cipuri, 158, 309
 random, serviciu, 69-70
 RARP (Reverse ARP), 454-455
 rata în buzi, 367-368, 401-402, 488
 RAWRITE.EXE, 38, 39
 rcmd.pl, 225-227
 RCPT TO:, comandă, 504-505
 RCS (Revision Control System), 58-59, 136, 578-584, 667
 rcs, comandă, 582
 rcsdiff, utilitar, 578-579, 582
 rcsmerge, utilitar, 578-579
 RD (Receive Data), semnal, 405-406, 421-422
 rdev, comandă, 79-80
 rdist, 667
 read (citire), mod, 280
 read, comandă, 211-212, 259, 263-264
 read-only (disponibil numai pentru citire), sistem de fișiere, 79-80
 README.agx, 306-307
 README.ati, 305
 README.cirrus, 305
 README.install, 526
 README.Oak, 306-307
 README.P9000, 306-307
 README.S3, 306-307
 README.S3V, 306-307
 README.trident, 305
 README.TXT, 30-31
 README.Video7, 305
 README.W32, 306-307
 README.WstDig, 305
 Real Audio, 494
 RealTek, 27-28, 305
 Red Hat, server FTP, 91, 95-96, 120-121
 Red Hat, sit Web, 45, 91
 redesenare, viteza de, 140
 regexp, comandă, 259, 260
 regsub, comandă, 259, 260
 reîmprospătare, viteza de, 59, 335, 336. *Vezi și* sincronizare verticală (viteza de reîmprospătare)
 reîncărcare. *Vezi și* încărcare (booting)
 cu discuri de boot, 32, 37
 după configurarea și instalarea LILO, 119-120
 după efectuarea modernizărilor, 123
 imediat după închidere, 90
 remedii (patches). *Vezi și* modernizări aplicare, 97-102
 descriere generală, 91, 97, 126
 exemple de, 99-101
 nucleu, 97-102
 numere pentru, 99, 126
 obținere, 99-101
 REN, comandă, 238-239
 rename, comandă, 259
 RENAME, comandă, 238-239

renunțare, acțiune, 163-164, 485-486
 repetare, întârziere la, 357, 358-360
 repetiție, operator de, 219-220
 Reset, buton, 52
 reset, comandă, 478
 RESOURCE_MANAGER, șir, 181-183
 RESTORRB.EXE, 37
 resurse (listate după nume). *Vezi și* resurse
 XmNaccelerators, 598-599
 XmNactivateCallback, 597
 XmNancestorSensitive, 598-599
 XmNbackground, 598-599
 XmNbackgroundPixmap, 598-599
 XmNborderColor, 598-599
 XmNborderWidth, 598-599
 XmNbottomShadowColor, 601-602
 XmNbottomShadowPixmap, 601-602
 XmNcolormap, 598-599
 XmNdepth, 598-599
 XmNdestroyCallback, 598-599
 XmNforeground, 601-602
 XmNheight, 598-599
 XmNhelpCallback, 601-602
 XmNhighlightColor, 601-602
 XmNhighlightOnEnter, 601-602
 XmNhighlightPixmap, 601-602
 XmNhighlightThickness, 601-602
 XmNmappedWhenManaged, 598-599
 XmNnavigationType, 601-602
 XmNscreen, 599-600
 XmNsensitive, 599-600
 XmNshadowThickness, 601-602
 XmNtopShadowColor, 601-602
 XmNtopShadowPixmap, 601-602
 XmNtranslations, 599-600
 XmNtraversalOn, 602-603
 XmNunitType, 601-602
 XmNuserData, 601-602
 XmNwidth, 598-599
 resurse. *Vezi și* resurse (listate după nume)
 descriere generală, 179-183
 fișiere, locație, 181-183
 nume, 180-182
 primitive (widget), 594-597
 standard, 182-183
 X Window System și, 179, 592-593, 594-597, 598-600

return, comandă, 259
 Rețea, nivel, 427-428, 428-429
 rețele dial-up, 137, 470-471, 554-555
 rețele locale (LAN), 635-636, 638-648. *Vezi și* LAN
 rețele. *Vezi și* LAN (Local Area Network)
 accesibilitate, verificare, 456
 adaptoare, descriere generală, 27-28
 configurare, 17, 65-67, 134-135, 459-460
 descriere generală, 130-134, 426-463
 instalare și, 17, 65-67, 69-70
 măști, 66-67, 434
 modernizări Linux și, 110
 monitorizare, 134-135
 nume de dispozitiv, 451-453
 operații de administrare, 134-135
 servicii, 69-70
 stare, verificare, 461-462
 testare, 456
 rezoluție
 imprimantă, 71-73
 monitor, 158, 164-165, 295-296
 pentru moduri video, 64-66, 80-81
 rf, câmp, 385
 RFC (Requests for Comments), 429-431, 470-471, 472, 524
 RGB, culori, 152, 174-175, 194-196, 204, 335, 352, 702. *Vezi și* culori
 RGB, fișier, 159-160, 160-161
 rhsound, pachet, 58
 RI (Ring Indicator), semnal, 405-406
 RIP (Routing Information Protocol), 69-70
 rlog, comandă, 582, 583
 rlog, utilitar, 578-579
 rlogin, 131-132
 rm, câmp, 385, 386-387
 rm, comandă, 202, 207-208
 rmdir, comandă, 208
 Rock Ridge Extensions, 336
 ROM (memorie read-only – numai pentru citire), 22-23, 280, 313-314
 root, (super)utilizatorul, 17, 74, 78, 98, 104
 comenzi mtools și, 237
 dischete și, 315-316
 DOSEMU și, 239-240
 închidere și, 322
 modemuri și, 414

montare sisteme de fișiere și, 232-233
redare CD-uri audio și, 344
rețele și, 455
schimbarea nivelurilor de rulare ca,
173-175
tipărire și, 381
route, comandă, 489-490, 493
routed, serviciu, 69-70
routing, 109-110, 434-436, 491-495
RowColumn, primitivă, 603-604
rp, câmp, 386-387
RPC (Remote Procedure Calls), 69-70,
127-128, 441-442
rpm, comandă, 92, 94-95, 96, 120, 133-134,
237, 241
RTS (Request to Send), semnal, 404, 407
rusersd, serviciu, 69-70
rwhod, serviciu, 69-70

S

/S, opțiune, 32
s, opțiune, 359-360
S, registru, 411-413
S3, plăci video, 307-309
salt la linie nouă (line-feed), caracter,
194-195, 250-251, 392-394, 412-413,
419-420, 486-487
Samba LAN Manager, 633-635, 639-645,
650. *Vezi și* LAN Manager
samba, pachet, 58-59
sample.txt, 244-245
Sanyo, unități de CD-ROM, 25-26, 337-338
Sash, funcție, 601-602
satview, 663-664
sb, câmp, 385
/sbin/lilo, program, 318-319
sc, câmp, 385
scale, comandă, 267-268
Scale, primitivă, 603-604
scan, comandă, 259
SCANDISK, 37
ScanHTMLTitles, opțiune, 534-535
scd, driver, 345
Scheifler, Robert W., 143
Schneider & Koch, plăci Ethernet, 28,
447-448
Schwartz, Randal, 215-216

SCO UNIX, 129-130
scripturi. *Vezi și* programe shell (interpretor)
descriere generală, 191, 213, 246-278
filtre de intrare, 386-387, 551-552
fișiere, 475-481
netcfg, 456-460
rețele dial-up și, 488-492
scriere, pentru tipărire cu smbclient,
552-554
startx, 162-164, 165, 298
toolhelp, 275-276
Xsession, 170-171
scrollbar, comandă, 267-268
ScrollBar, funcție, 601-602
ScrolledWindow, primitivă, 603-604
SCSI (Small Computer Systems Interface)
blocare, 333
bune de dispozitiv, 49, 315-316
controllere, listă de, 323
descriere generală, 23-25, 312-314, 322
drivere, încărcare, 47-48
instalare și, 21-22, 23-26, 28-29, 47-48,
49
IRQ și, 289-290
modernizări Linux și, 91, 102, 108-109,
110-112, 113, 118, 122
partiții și, 236, 316-317
plăci adaptor, 468
rezolvarea problemelor, 329
unități de CD-ROM, 25-26, 337, 340-341
SCSISEL.EXE, 356
Seagate, 24, 324, 327
Sealevel Systems, plăci seriale, 423
sectoare, 313-315
securid, comandă, 478
securidf, comandă, 478
securitate. *Vezi și* parole
FTP anonim și, 514-516
obținerea de informații despre, 515-517
parafocuri (firewalls), 109-110, 454-455,
493, 544
pentru directoare, 205-207
sed, utilitar, 128-129
seek, comandă, 259
SEEQ, plăci Ethernet, 28, 447-448
selectarea componentelor de instalat, fază de
instalare, 44, 55-60

selection, comandă, 268-269
SelectionBox, primitivă, 603-604
selecția intrării, 141-142
semiduplex, mod 464
semn de exclamare (!), 214-215, 221
semnale
semnalul CTS (Clear to Send), 404, 407,
417-419, 487
semnalul DSR (Data Set Ready),
405-406
semnalul DTR (Data Terminal Ready),
405-406
semnalul RD (Receive Data), 405-406,
421-422
semnalul RI (Ring Indicator), 405-406
semnalul RTS (Request to Send), 404,
407
semnalul SIGKILL, 197-198
semnalul Signal Ground, 405-406
semnalul TD (Transmit Data), 405-406,
421-422
semnul diez (#), 99, 104-105, 159-160,
172-173, 214-215, 216-217, 238, 241,
251-252, 300, 375-376, 382-383, 443-444,
521, 531
semnul dolar (\$), 211-212, 216-217, 221,
249-250, 250-251, 252, 566-567, 580-581
semnul egal (=), 361
semnul întrebării (?), 198-199, 521
semnul minus (-), 148
semnul plus (+), 74, 361, 412-413
semnul procent (%), 216-217, 218, 247, 567
send, comandă, 268-269, 478, 480-481
sendmail, 69-70, 500-504, 667
sendmail, serviciu, 69-70
sensibilitate la litere mari/mici, 233-234
Separator, funcție, 601-602
Serial Line Internet Protocol (SLIP), 426,
451-452, 471-473, 553-554, 554-555
comprimat (CSLIP), 472-473, 474-475,
478
configurare, 134-135
dedicat, 137
descriere generală, 131-133
modemuri și, 413
modernizări Linux și, 112

serve de baze de date, 141
serve de fișiere, 141
serve de nume, 437-438
serve independente, 444-445
serve Web, 542-543, 544-546
Apache HTTP, server, 58-59, 69, 95-96,
518-519, 522-523, 528-529, 541
descriere generală, 131-132, 132-133
la sediul clientului, 545-546
setare, 527-537
serve. *Vezi și* serve Web; serverX
Apache HTTP, server, 58-59, 69, 95-96,
518-519, 522-523, 528-529, 541
descriere generală, 131-132, 132-133
Gopher Server, 58-59
inetd, server, 69, 442-444
nume de serve, 437-438
serve de baze de date, 141
serve de fișiere, 141
serve independente, 444-445
ServerFlags, 159-160, 160-161
servicii
configurare, 67-70
serviciul atd, 69
serviciul crond, 69
serviciul gpm, 69
serviciul innd, 69
serviciul kernel, 69
serviciul keytable, 69
serviciul lpd, 69
serviciul mars-nwe, 69
serviciul named, 69
serviciul nfs, 69-70
serviciul portmap, 69-70
serviciul pstgresql, 69-70
serviciul random, 69-70
serviciul routed, 69-70
serviciul rusersd, 69-70
serviciul rwhod, 69-70
serviciul sendmail, 69-70
serviciul smb, 69-70
serviciul snmpd, 69-70
serviciul sound, 69-70
serviciul syslog, 69-70
serviciul ypbind, 69-70
Sesiune, nivel, 427-428

set, comandă, 249-250, 259
 setleds, 363-364
 setserial, 417-419
 seturi de instrumente, 141-142, 193-194, 591-592
 sf, câmp, 385, 386, 387-388
 sh, câmp, 385, 386, 387-388
 SHELL, variabilă de mediu, 196
 shift, funcție
 showkey, comandă, 361
 shutdown, comandă, 32, 79, 145, 368. *Vezi și* închidere
 SIGKILL, semnal, 197-198
 sigle, modele de fundal pentru, 177-178
 Sigma LaserView PLUS, plăci video, 21
 Signal Ground, semnal, 405-406
 sigur, FTP anonim, 514-516
 SIIG, plăci seriale, 423
 sin, funcție, 253
 sincronizare verticală, (viteze de reîmprospătare), 59, 61, 157, 161, 176, 342, 344. *Vezi și* reîmprospătare, viteza de sincronizare a frecvențelor
 pe orizontală, 62-63, 143-144, 148, 161, 301-302, 303
 pe verticală, 63, 64-65, 144-145, 148, 161, 301-302, 303
 Sis, plăci video, 305
 sistem aflat la distanță, obținere de informații, 474-476
 sisteme de dirijare a știrilor (BBS), 132-134, 399
 situri Web
 situl Web Caldera, 539
 situl Web CERT (Computer Emergency Response Team), 515-516
 situl Web Metro Link, 309-310
 situl Web PASC, 128-129
 situl Web Red Hat, 45, 91
 situl Web Xi Graphics, 309-310
 skey, comandă, 478
 slash (/), 44, 48-49, 202-203, 221, 231-232, 233-234, 270, 510-511, 642-643, 648
 slash (/), 44, 48-49, 202-203, 221, 231-232, 233-234, 270, 510-511, 642-643, 648
 sleep, comandă, 478
 SLIP (Serial Line Internet Protocol), 426, 451-452, 471-473, 553-554, 554-555
 comprimat (CSLIP), 472-473, 474-475, 478
 configurare, 134-135
 dedicat, 137
 descriere generală, 131-133
 modemuri și, 413
 modernizări Linux și, 112
 Small Computer Systems Interface (SCSI)
 blocare, 333
 controllere, listă, 323
 descriere generală, 23-25, 312-314, 323
 drive, încărcare, 47-48
 instalare și, 21-22, 23-26, 28-29, 47-48, 49
 IRQ și, 289-290
 modernizări Linux și, 91, 107, 108-109, 110-112, 113, 118, 122
 nume de dispozitiv, 49, 315-316
 partiții și, 236, 316-317
 plăci adaptor, 468
 rezolvarea problemelor, 329
 unități de CD-ROM, 25-26, 337, 340-341
 SMB (Samba) Connectivity, 58-59, 69-70, 549-554
 SMB (Server Message Block), 545-546
 smb, serviciu, 69-70
 smbclient, 549-554
 smbfs, pachet, 58-59
 smbprint.tcl, 552-554
 SMC, plăci Ethernet, 28, 447-448
 SMP (Symmetric Multi Processing), versiune de Linux, 115-116
 SMTP (Simple Mail Transfer Protocol), 429, 501-502, 504-506
 sndconfig, pachet, 58
 SNMP (Simple Network Management Protocol), 58-59, 69-70, 429, 439-440
 snmpd, serviciu, 69-70
 Socket API, 438-439, 440-441
 socluri, 438-439, 440-441
 socluri de flux, 439
 software, dezvoltare, 58-59, 135-136, 555, 557-590
 Solaris, 129-130
 solid, opțiune, 175-176

SONET/SDH (Synchronous Optical Network/Synchronous Digital Hierarchy), 473
 Sony, unități de CD-ROM, 25-26, 337-338, 346-347
 sonycd535, driver, 346-347
 Sound Blaster, 24-27, 192-193, 324, 339-340, 342, 343-344, 347-348, 349-351, 355-356
 Sound Galaxy, 27-28
 sound, serviciu, 69-70
 source, comandă, 259
 SPARC, procesoare, 280-281
 spațiu de schimb, 29-30, 34-35, 53-55
 spațiu, caracter, 419-420
 speed, comandă, 478
 split, comandă, 259, 260, 261, 263-264
 Sprint, 553-554
 sqrt, funcție, 253
 SRAM, 468
 SRI (Stanford Research Institute), 435-436
 st, câmp, 385
 standardul POSIX.1b, 127-128
 standardul POSIX.1c, 127-128
 standardul POSIX.1d, 127-128
 standardul POSIX.1e, 127-128
 standardul POSIX.1f, 127-128
 standardul POSIX.1g, 127-128
 standardul POSIX.1h, 127-128
 standardul POSIX.2, 128-129, 129, 136
 standardul POSIX.5b, 128-129
 StarOffice Suite, 539, 540
 START, biți de, 401-402
 Start, buton, 82
 Start, meniu, 40, 187
 startx, comandă, 168, 196-197, 369-370
 startx, script, 162-164, 165, 298
 status, fișier, 381-382
 stații de lucru, 14, 58-59, 135, 169-170, 280-281, 542-544
 STB-4COM, plăci seriale, 423
 stderr, cuvânt-cheie, 262-263
 STDERR, handle de fișier, 227-228
 stdin, cuvânt-cheie, 262-263
 STDIN, handle de fișier, 227-228
 stdout, cuvânt-cheie, 262-263
 Stick/Unstick, opțiune, 185-187
 stil specific, 141-142
 STOP, biți de, 400-401, 401-402
 stopbits, comandă, 478
 string, comandă, 260, 261
 structuri de control, 212-213
 stty, comandă, 417-419, 419-420
 sub-expect, secvențe, 484-486
 subrețele, 434
 subrutine, 229
 substituții, 249-251
 Sun Microsystems, stații de lucru, 135, 280-281, 544
 sunet analogic, 348-349
 Microsoft, sistem de, 26-27, 291-292, 348-349, 350-351
 pe calculatoare laptop, 291-292
 sunet. *Vezi și* plăci de sunet
 sunset.gif, 178-179
 Super VGA (Video Graphic Array), ecrane, 20, 291-292, 299
 superutilizatori (root), 17, 74, 78, 98, 104
 comenzi mtools și, 237
 dischete și, 315-316
 DOSEMU și, 239-240
 închidere și, 322
 modemuri și, 414
 montarea sistemelor de fișiere și, 232-233
 redare CD-uri audio și, 344
 rețele și, 455
 schimbarea nivelurilor de rulare la, 173-175
 tipărire și, 381
 suport pentru animație, pachete, 57
 suprafață de lucru
 stiluri, 188-190
 virtuală, 185-190
 suprafață de lucru virtuală, 185-190
 SuppressDescription, opțiune, 534-535
 SuppressLastModified, opțiune, 534-535
 SuppressSize, opțiune, 534-535
 SVGA (Super VGA), ecrane, 20, 291-292, 299
 swap, partiții, 17, 48, 50-51
 switch, comandă, 256-257, 260
 switch, instrucțiuni, 253-257
 SymLinksWithOwnerMatch, opțiune, 536-537

SYN flooding, atacuri, 109-110
 SYS.COM, 241, 243
 syslog, serviciu, 69-70
 System V (UNIX), 106-107, 113-114, 126, 129-130
 system, funcție, 128-129, 225-227

S

șiruri de caractere, 181-182, 182-183, 260-262
 ștergere
 fișiere, 238
 partiții, 34-35, 52, 52-53
 text, 656-658

T

t, opțiune, 462, 476, 480-481
 Tab, tastă, 43, 78
 tabele de conversie, 599-600
 tabele de dispersie, 262
 tabulator vertical (v), caracter, 250-251
 tabulator, caracter, 419-420, 486-487
 tan, funcție, 253
 tar, fișiere, 93
 tar, utilitar, 93, 133-134, 134-135
 tastaturi. *Vezi și* apăsări de taste
 configurație, 358-359
 descriere generală, 357-365
 evenimente, 276-277, 611, 612-613
 hărți, 69, 358, 359-361
 instalare și, 22-23, 46, 69
 întârziere la repetarea unui caracter, 357, 358-360
 rata de repetare, 357, 358-360
 terminologie/notații, 358-359
 tipuri de, selectare, în timpul instalării, 46
 X Window System și, 147, 159-160
 taste cu săgeți, navigare cu, 43, 46, 56
 TCL (Tool Command Language). *Vezi și* Tcl/Tk
 comenzi încorporate, 258-260
 descriere generală, 248-264
 manipularea șirurilor cu, 260-262
 Tcl/Tk, 52, 147, 783. *Vezi și* TCL (Tool Command Language)
 descriere generală, 246-278
 surse de informare despre, 247-248

TCP (Transmission Control Protocol), 507, 508, 510. *Vezi și* TCP/IP (Transmission Control Protocol/Internet Protocol)
 TCP/IP (Transmission Control Protocol/Internet Protocol), 540, 541-542, 545-546
 configurarea calculatorului-gazdă și, 497
 descriere generală, 130-131, 131-132, 426, 429
 diagnosticare, 460-462
 instalare și, 44, 58-59, 65-67
 instalare/configurare, 44, 65-67, 452-460
 Internet și, 429-438
 modele de rețea, 428-429
 modernizări Linux și, 109-110, 110
 pachete, 58-59
 rețele dial-up și, 470-471
 routing, 434-436, 493, 494
 servicii, 437-444
 surse de informare despre, 429
 UUCP și, 134-135
 TD (Transmit Data), semnal, 405-406, 421-422
 Teac, unități de CD-ROM, 25-26
 TearOffButton, funcție, 601-602
 Tekram, 24, 324
 Telix, 421-422
 tell, comandă, 260
 telnet, 58-59, 131-132, 429, 441-442, 471, 499, 503-506, 521, 522-523, 529
 term, comandă, 478, 479-480
 TERM, variabilă de mediu, 168-169, 196
 terminale, 399, 414-418, 421-423, 554-555, 645
 terminale virtuale, 197-199
 terminator, probleme cu, 323-324
 test, comandă, 212-213
 test, program, 209-210
 testare, 209-210, 212-213
 discuri de boot (încărcare), 32
 fișier de configurare Samba, 548-552
 instalare modem, 421-422
 pagini man, 665-666
 testparm, 548-549
 TeX, 387-388, 541, 667
 Texas Instruments, 349
 text, comandă, 267-268

Text, funcție, 601-602
 text. *Vezi și* fonturi
 desenare, 614-616, 621-622
 modul input (introducere text), 641-642, 647
 prelucrare, 640-666
 ștergere, 656-658
 tf, câmp, 385
 TFTP (Trivial File Transfer Protocol), 441-442
 theCanvas, variabilă, 257
 ThunderBoard, plăci de sunet, 27-28, 349
 TIFF, imagini, 178-179, 394-395
 tilda (~), 205, 221, 647
 time, comandă, 260
 timeout, comandă, 479
 tipărire. *Vezi și* imprimante
 cu efecte speciale, 375-376
 cu smbclient, 552-554
 descriere generală, 372-398
 din fundal, 376-388
 din punctul de vedere al utilizatorului, 373-385
 probleme/soluții, 391-397
 verificarea cozilor de tipărire, 374-375
 tk, comandă, 268-269
 tkerror, comandă, 268-269
 Token Ring, rețele, 130-131, 428-429, 468
 token-ring, protocol, 130-131
 toolhelp, script, 275-276
 top, utilitar, 134-135
 topcx, fișier, 193-194, 194-195
 toplevel, comandă, 267-268
 TopLevelShell, primitivă, 599-601
 tr, câmp, 385
 tr, comandă, 194-195
 tr, utilitar, 128-129
 trace, comandă, 260
 trackball, 409, 425. *Vezi și* dispozitive de indicare
 TransientShell, primitivă, 599-601
 Transmission Control Protocol (TCP), 507, 508, 510. *Vezi și* TCP/IP (Transmission Control Protocol/Internet Protocol)
 Transmission Control Protocol/Internet Protocol (TCP/IP), 540, 541-542, 545-546
 configurarea calculatorului-gazdă și, 497

descriere generală, 130-131, 131-132, 426, 429
 diagnosticare, 460-462
 instalare și, 44, 58-59, 65-67
 instalare/configurare, 44, 65-67, 452-460
 Internet și, 429-438
 modele de rețea, 428-429
 modernizări Linux și, 109-110, 110
 pachete, 58-59
 rețele dial-up și, 470-471
 routing, 434-436, 493, 494
 servicii, 437-444
 surse de informare despre, 429
 UUCP și, 134-135
 Transport, nivel, 427-428, 428-429
 Trantor, 24, 324, 328
 Trident, plăci video, 304-305
 trn, pachet, 57, 667
 troff, 387-388, 660-661
 Tseng, plăci video, 304-305, 305-306
 TSR (Terminate and Stay Resident Programs), 105-106, 129
 tub catodic, 140
 Turtle Beach, plăci de sunet, 27-28, 349
 Twincom, plăci seriale, 423
 TYPE, comandă, 238-239
 typedef, instrucțiune, 594-596

U
 u, opțiune, 582
 U.S. Robotics, 413
 UART (Universal Asynchronous Receiver/Transmitters), 399-401, 403-404, 422-423
 UDP (User Datagram Protocol), 429, 439-440
 ugetty, 417-419
 UltraStor, 24, 324, 328
 uname, comandă, 98
 unități de bandă magnetică, 107-108, 115-116
 unități de CD-ROM
 acceptate de Linux, 336-337
 brevetate, 337-338
 descriere generală, 24-27
 modernizări Linux și, 108-109

parametri de boot, 342-343
 rezolvarea problemelor, 338-340
 unități de disc. *Vezi și* unități de dischete
 concepte, 313-315
 descriere generală, 312-334
 nume de dispozitiv pentru, 315
 probleme specifice pentru Linux, 319-323
 unități de dischetă
 comparație cu unitățile de CD-ROM, 24
 descriere generală, 315-316
 formatare, 238-240
 instalare și, 22-23, 24, 29-30, 48-49
 modernizări Linux și, 107-108, 115-116
 montare, 233-235, 316
 MS-DOS, 233-235, 238-240
 nume, 48-49
 unități de hard-disc. *Vezi și* partiționare
 defragmentare, 36-37
 descriere generală, 22-23
 imagini, inițializare, 242-244
 înregistrări de siguranță, 31
 litere pentru, 28-29, 48
 secundare, 22, 28-29
 Universitatea Berkeley, California, 246, 247
 Universitatea din Helsinki, 14-15
 UNIX
 -to-UNIX Copy (UUCP), 132-134, 500-501, 505-506
 BSD (Berkeley Software Distribution), 113-114, 126, 129, 373-374, 377-378, 391
 nume de fișiere, 48
 pagini man, 59, 84-85, 660-666
 platformă, Linux ca, 126-130
 System Laboratories (USL), 584
 System V; 106-107, 113-114, 126, 129-130
 unknown, comandă, 260
 unmount, comandă, 232-233, 235, 236
 unset, comandă, 260
 unzip, pachet, 57
 update, comandă, 268-269
 uplevel, comandă, 260
 UPS (Uninterruptible Power Supply), 649.
Vezi și alimentarea cu tensiune
 upvar, comandă, 260

URL (Uniform Resource Locators), 519-521
 USENET, grupuri de discuții, 77, 101, 144, 589-599. *Vezi și* grupuri de discuții
 User Configurator, instrument, 82-83
 USL (UNIX System Laboratories), 584
 utilitare gratuite, 134-135
 utilizator(i). *Vezi și* utilizatori, root/super
 adăugare/eliminare, 133-134
 conturi, adăugare, 82-83
 nume, 531
 utilizatori, root/super, 17, 74, 78, 98, 104
 comenzi mtools și, 237
 dischete și, 315-316
 DOSEMU și, 239-240
 închidere și, 322
 modemuri și, 414
 montarea sistemelor de fișiere și, 232-233
 redare CD-uri audio și, 344
 rețele și, 455
 schimbarea nivelurilor de rulare, 173-175
 tipărire și, 381
 UTM (Uniform Transfer Model), 600-601
 UTP, cablu, 445-446
 UUCP (UNIX-to-UNIX Copy), 132-134, 500-501, 505-506
 uugetty, program, 417-420, 420-422

V

V, comandă, 409
 values, funcție, 218
 variabile (listate după nume). *Vezi și* variabile
 variabila count, 249-251
 variabila DISPLAY, 196
 variabila HOME, 168-169, 196
 variabila LOGNAME, 196
 variabila NNTPSERVER, 506-507, 507-508, 509
 variabila ONBOOT, 459-460
 variabila PATH, 146-147, 193-194, 195-196, 211-212, 213-214, 218, 247, 262-263, 265
 variabila PRINTER, 380
 variabila PS1, 196
 variabila SHELL, 196
 variabila TERM, 168-169, 196
 variabila theCanvas, 257
 variabila XENVIRONMENT, 181-182

variabile de mediu. *Vezi și* variabile de mediu
 (listate după nume)
 descriere generală, 168-169
 interpretor Bash și, 195-196, 210-212
 Tcl/Tk, script-uri și, 262-263
 variabile globale, 257
 variabile. *Vezi și* variabile (listate după nume)
 de tip masiv (tablou), 216-217
 dip, 478-479
 make, utilitar și, 566-567, 570-571
 Perl și, 215-219
 predefinite, 218-219
 scalare, 216-217
 substituția, 250-251
 Tcl/Tk, programare și, 248-249
 verbose, parametru, 179
 versiuni. *Vezi și* remedii (patches)
 control, 578-584
 de Motif, 600-601
 descriere generală, 126
 numere, major/minor, 97-102
 VESA Local Bus (VLB), 19, 282
 vf, câmp, 385
 VFAT, sistem de fișiere, 113-114
 VGA (Video Graphic Array), ecrane, 21-22, 162, 291-292, 297-298
 vi, editor de text, 128-129, 195, 200-201, 211-212, 214-215, 513, 557-558, 640-641, 645-651, 652-653
 Video 7, plăci video, 21, 305
 Vim, editor de text, 668-669
 Viper PCI, plăci video, 20
 viteze de repetare, 357, 358-360
 VLB, magistrală, 19, 282
 vmlinuz, 79-80
 VRAM (video RAM), 157, 334-336. *Vezi și*
 memorie video
 VT100, clasă, 180-181

W

WAIS (Wide Area Information Service), 441-442
 wait, comandă, 479
 Wall, Larry, 213, 214, 215-216
 WAN (Wide Area Networks), 131-132
 watchdog, dispozitiv de semnalizare, 115-116

WaveBlaster, plăci de sunet, 27-28
 Weitek, plăci video, 310
 Welcome, casetă de dialog, 61-62
 Western Digital, 24, 291-292, 304-307, 324, 329, 447-448
 which, comandă, 213
 while, ciclu, 222-225
 while, comandă, 253-257, 260
 while, instrucțiune, 215-216, 223-224, 253-257
 while, structură de control, 212-213
 Windows 3.x (Microsoft), 14-15, 18, 28-31
 Windows 95 (Microsoft)
 bara de operații, 187
 cursoare și, 178-179
 FDISK și, 32
 folosit în mediu de afaceri, 540, 545-546, 549
 instalare și, 18, 21-22, 28-31, 32, 40-41, 70-71, 82
 în rețele dial-up, 494-495
 LILO și, 317-318
 partiționare hard-disc sub, 28-31
 suprafața de lucru, 185-190
 Windows NT (Microsoft)
 folosit în mediu de afaceri, 541, 553-554
 instalare și, 18, 70-71
 standarde POSIX și, 127-128
 WINE, proiect, 135
 winfo, comandă, 268-269
 wish, interpretor, 265-267
 WM Style, opțiune, 188-189
 wm, comandă, 268-269
 Word (Microsoft), 541-542, 545
 WordPerfect (Corel), 129, 135, 539, 540, 545
 World Wide Web. *Vezi și* browsere de Web;
 pagini Web; servere Web
 adrese URL și, 519-521
 navigarea prin, 524-528
 pagini, 518-519, 526-527, 545-546
 publicarea în, 136
 Wyse V/386 UNIX, 129-130

X

x opțiune, 196-197
 X protocol, 139, 141
 X server
 comercial, 309-310

configurare, 144-146, 160-161, 162
 descriere generală, 20, 139
 moduri video și, 163-165
 oprire, 163-164
 pachet, 57
 resurse, 608-609
 selectare, 157-158, 159-160, 297-298
 tastaturi și, 362-363
 X Window System. *Vezi* și X, server;
 XFree86
 configurare, 143-163
 conturi de utilizator, 82-83
 descriere generală, 129-131, 139-167
 ieșire din, 83-84
 informații despre, 277
 instalare și, 14, 20, 21-22, 28-29, 44,
 57-59, 80-84
 istoric, 141-143
 mouse și, 364, 366-370
 pachet, 59
 personalizare, 175-185
 pomire și, 80-82, 168-169, 169-186
 programare, 591-693
 proiectul WINE și, 135
 rulare, 162-165
 tastaturi și, 358-359, 361-364
 Tcl/Tk, programare și, 264, 277
 terminale virtuale și, 198-199
 X.25, rețele, 473
 xanim, 667
 xbill, pachet, 57
 xboard, pachet, 57
 xboing, pachet, 57
 xchomp, pachet, 57
 Xclients, fișier, 175
 XColor, structură, 631-632
 Xconfigurator, 17, 61-66
 xconsole, 169-170
 xdemineur, pachet, 57
 xdm (X Display Manager), 169-171, 174
 xdm-config, fișier, 169-170, 170-171
 XENVIRONMENT, variabilă de mediu,
 181-182
 Xerox, 445
 xevil, pachet, 57
 XF86Config, fișier, 65-66, 145-165, 198-199,
 297-306, 361-364, 367-368

xf86config, program, 145-160, 161-162, 298,
 300-303
 xfig, 683-684
 xfishtank, pachet, 57
 xfm, 667, 676-677
 XFree86, 9, 151. *Vezi* și X Window System
 calculatoare laptop și, 289-292
 configurare, 143-160
 descriere generală, 129-131, 143
 dispozitive de indicare și, 357
 instalare și, 21-22
 pachete, 57
 plăci video și, 283, 306-307
 servere X comerciale pentru, 309-310
 xgalaga, pachet, 57
 xgammon, pachet, 57
 xgopher, pachet, 57
 Xi Graphics, sit Web, 309-310
 xinit, 170-171
 .xinitrc, fișier, 163, 170-171
 XIP (Execute in Place), 466-467
 Xircom, 448-449, 469
 xjewel, pachet, 57
 xlander, pachet, 57
 Xlib, bibliotecă, 264, 591, 603-638
 caracteristici comune, 606-609
 o trecere în revistă a funcțiilor, 605-608
 XmDrawingArea, primitivă, 597-598
 xmlines.c, 624-629
 XmNaccelerators, resursă, 598-599
 XmNactivateCallback, resursă, 597
 XmNancestorSensitive, resursă, 598-599
 XmNbackground, resursă, 598-599
 XmNbackgroundPixmap, resursă, 598-599
 XmNborderColor, resursă, 598-599
 XmNborderWidth, resursă, 598-599
 XmNbottomShadowColor, resursă, 601-602
 XmNbottomShadowPixmap, resursă,
 601-602
 XmNcolormap, resursă, 598-599
 XmNdepth, resursă, 598-599
 XmNdestroyCallback, resursă, 598-599
 XmNforeground, resursă, 601-602
 XmNheight, resursă, 598-599
 XmNhelpCallback, resursă, 601-602
 XmNhighlightColor, resursă, 601-602
 XmNhighlightOnEnter, resursă, 601-602

XmNhighlightPixmap, resursă, 601-602
 XmNhighlightThickness, resursă, 601-602
 XmNmappedWhenManaged, resursă,
 598-599
 XmNnavigationType, resursă, 601-602
 XmNscreen, resursă, 599-600
 XmNsensitive, resursă, 599-600
 XmNshadowThickness, resursă, 601-602
 XmNtopShadowColor, resursă, 601-602
 XmNtopShadowPixmap, resursă, 601-602
 XmNtranslations, resursă, 599-600
 XmNtraversalOn, resursă, 602-603
 XmNunitType, resursă, 601-602
 XmNuserData, resursă, 601-602
 XmNwidth, resursă, 598-599
 xmodmap, 360-361, 362-363
 xmodtest, 362-364
 xmqquit, 593-595
 xmutil.c, 582-583
 XNS (Xerox Network System), 454-455
 XOFF, caracter, 407
 XON, caracter, 407
 xpaint, 130-131, 667, 682-683
 xpat2, pachet, 57
 xpilot, pachet, 57
 xplayed, program, 343-344
 xprop, utilitar, 181-183
 xpuzzles, pachet, 57
 Xsession, script, 170-171
 xsetroot, program, 176-179
 xsnow, pachet, 57
 XtAddCallback, funcție, 597
 XtAddEventHandler, funcție, 598
 XtAllCallback, funcție, 592-593
 XtAppInitialize, funcție, 592-593
 XtAppMainLoop, funcție, 592-593

XtCreateManagedWidget, funcție, 595-596,
 596-597
 XtDestroyWidget, funcție, 598-599
 xterm, 163, 170-171, 175-176, 179-185, 191,
 195-199, 202, 244-245, 265, 369-370,
 375-376, 414, 526, 594-595, 651-652,
 661-662
 XtRealizeWidget, funcție, 592-593
 xtrojka, pachet, 57
 XtSetArg, macro, 595-596
 XtSetMappedWhenManaged, funcție,
 598-599
 XtVaCreateManagedWidget, funcție,
 592-593, 596-597
 xv, program, 130-131, 178-179, 667, 679-680
 Xview, 591
 XWD (X Window Dump), format, 193-194,
 194-195, 197-198
 xwdtopnm, program, 193-194
 xxgdb, program de depanare, 557-558

Y

yacc, 667
 Yamaha, placă de sunet, 349
 ypbind, serviciu, 69-70

Z

zcat, comandă, 85
 Zenith, plăci Ethernet, 28, 447-448
 zImage, 117, 118
 zip, pachet, 57
 Zip, unități (Iomega), 107-108, 328-329
 Znyx, plăci Ethernet, 28
 ZV (Zoomed Video), 466-467
 Zynx, plăci Ethernet, 447-448

